NOTHING you develop
will ever SUCCEED if
you can’t
COMMUNICATE
about it

Wouter de Vos

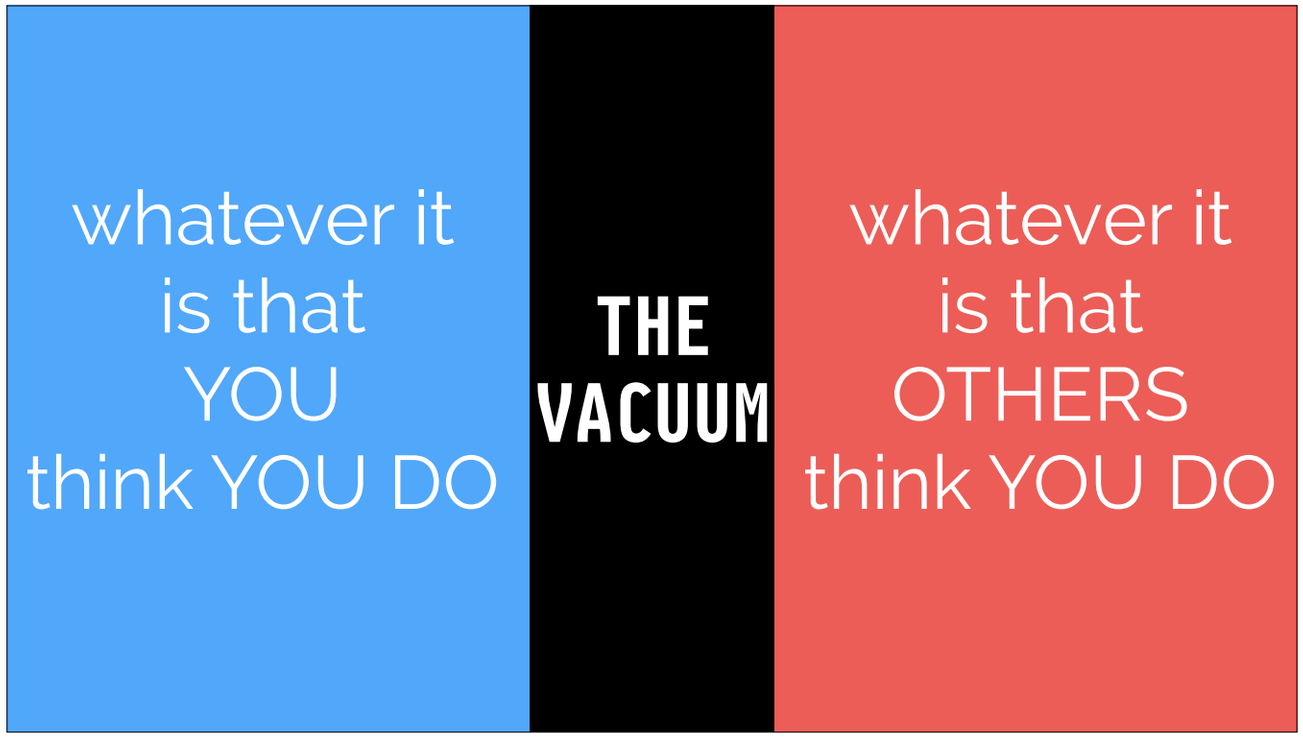
@wrdevos

THE VACUUM

The knowledge gap that needs to be filled with communication

whatever it
is that
YOU
think YOU DO

THE
VACUUM



Others have no idea what you do unless you TELL THEM

You have no idea what others do unless you ASK THEM

your BOSS
CUSTOMERS
colleagues
etc.

the “others”

Can you
EXPLAIN to your
MOTHER what you do
as a DEVELOPER?

What is development?

What does a dev need to do their job?

Have you ever built
SOMETHING that
was NOT what your
customer WANTED?

Have you ever built
SOMETHING that
was EXACTLY what your
customer WANTED?

COMMON
language

Knowledge gap: tech vs domain

Language ambiguity

cultural differences

USER STORIES tickets

REQUIREMENTS

meetings MOCKUPS

tests CODE REVIEWS

these are not just to
TELL you WHAT TO DO
but also to tell them
what you NEED to GET IT
DONE

create
AWARENESS

what is it that you do?

why is that important? what will it cost?

how is that for them?

create
SCOPE

What are you going to do?

What are you NOT going to do?

Laser focus.

create

COMMITMENT

Agree on what it is that you are doing – and what not.

What do you expect from THEM?

Have commitment on both ends.

get rid off
AMBIGUITY

Bottom line.

Be ultra clear

Expect that from them as well.

we are

THE EXPERTS

no other communication is this specific (except maybe law)

Only you know what you need

Educate your customers and colleagues

user story example

As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

Agile approach.

Less words but should be clear.

Is it?

requirements example 1

The user interface will show a list and a directory picker. The user can pick any number of directories to fill the list with directories that the user thinks do not need to be backed up. All directories not in the list will be backed up by default.
Rationale: The user should be able to exclude directories from the backup to save backup space.

Same story. More words.

Less ambiguity?

requirements example 2

The user interface will show a list of folders available for backup with a checkbox that can be toggled.
Rationale: The user should be able to choose which folders are, and which ones are not, backed up.

Same story. Interpreted wrong?

tough scenario's

CONSULTING CO.

get in the room with your customer and train them

Clarifying features is hard work.

Make sure they are willing to spend time on it

tough scenario's

OUTSOURCING

distance, languages, cultures

Find a structured language to communicate in (e.g. requirements)

Natural language might be too open to interpretation

tough scenario’s

PRODUCT CO.

be your own customer and
let dev do support shifts

Growing company can create vacuum because devs will move away more from the actual user (support).

tough scenario’s

SHARE