Scoping problem and designing solution

Chin, Chen

2023.12.05

Agenda

- intro - Why do we need to be scoped?
- main - Scoping problem & Scoping solution by design
- stories - cases * 3

Why do we need to be scoped?

What will happen if we do not scope the problem?

The results on the iceberg

- delivery delay / budget overrun
- current output cannot be monetized as product features
- stakeholders' expectations cannot be met
- project execution team morale and sense of achievement are low

The results under the iceberg

Inability to assess and calculate risks leads to insufficient information to make product or technical decisions, which in turn leads to strategic failure, and no matter how successful the implementation is, it cannot be compensated.

For example 🌹 on 💩, it never becomes a garden.

After this failure

Company might not invest in similar projects again.

Team might not dare to try similar requirements again.

Individual might not want to use similar technology again.

= lose-lose-lose

How to avoid this situation?

To get the right thing done, we need to do the right thing.

Scoping problem

What is the problem?

The situation would be the problem that the project is trying to solve.

Before we start to solve the problem,

we need to know **what the problem is.**

No offense, but...

usually the problem is not what the stakeholders say it is.

Why is that?

It might be because the stakeholders are not the users.

It might be because the stakeholders do not provide the full information.

Therefore, we have to do some research.

The definition of the problem

Why does the problem exist?

Why is this problem important?

Why do we need to solve this problem?

The goal of the project

Definite your project goal.

Emphasize the core problem you want to solve.

(Make sure everyone is on the same page.)

Scoping solution by design

The complexity of the problem

Analyze the complexity of the problem.

Split the problem into manageable parts.

Sub-problems have their own goal and sub-project.

Scoping Problem

Split the problem to reduce the complexity.

Which means

- Faster delivery
- Easily providing better solutions

  ⚠️ Because we all know that the clean code looks like, but the bad code usually has the diverse looks.
- Probably doing less jobs to get thing done.

Budget Concept

- Project management is consisted by time, scope and resource.
- When we can split the problem and solve it,
  we can control the risk.
- The biggest risk in the project is the uncertainty.

The uncertainty

- The uncertainty is the risk that we cannot control.
- The uncertainty is the time that we cannot estimate.
- The uncertainty is the situation that we cannot predict.

Case 1

The story is from my best friend.

She's a FE in a dutch company.

Someone (maybe is CEO or CFO) requested everyone to provide the idea to *"raise the revenue"*.

If I were her, I would ask...

- Is the goal to raise the revenue or to raise the profit?
- Who is the person asking for the idea?
- What is the current situation?
- How many resource can we use to find the answer?

Case 2

The story is from my previous company.

I were a Web Team Lead and I was dealing with the core feature of the product.

The product manager asked me if I can do some bug fixing.

And My design lead asked me to revise the UI to enhance UX.

If I could go back to the past, I would say...

- Let me revise the critical bugs first.
- Then I will refactor them to TS. (non functional requirement)
- After that I have time, I will revise the UI to enhance UX. (non functional requirement after mine.)

Case 3

My current core project is to split the current website to several services.
I did the EVERY SINGLE Technical decision correctly,
but not the Best One for the project in current phase.

Without blaming anyone, I would say...

- Let's clarify the current goal of the project.
  - And we basically need to split the website to several services.
  - We need to do it in a short time.
- The non functional requirements like upgrading critical dependencies, security, etc. are not the priority.

Thus We have to...

- We need to design the solution with the current resource.
  - We use the current source code via submodule.
  - We scope the current delivery is architecture design and traffic control.
  - We do the refactoring the core mass global items in the next phase.

Conclusion

We all know the best solution, but we cannot always do the best solution.

Hope we solve the problem by scoping it and designing the solution properly.

Questions?

Thank you