Software Systems Architecture

FEUP-M.EIC-ASSO-2023-2024

Ademar Aguiar, Neil Harrison

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Architecture, Ambiguity, and Abstraction

Let's talk about Specifications ...

Let's play fantasy developer!

You are on a team writing a new payroll management system for the company.

In small groups, discuss

- In an ideal world, what does the spec look like?
  - General characteristics
  - Give several examples
  - (write some notes)

If you are the architect ...

What does the spec look like (real world)?

“We need a new payroll management system”

Why?

It’s the nature of architecture: very little has been decided yet.

If you are the architect …
and you want a detailed spec:

The nature of architecture is ambiguity

Practical ramifications for this class:

- Assignments are ambiguous
- Your solutions will vary (from each other)
- Your solutions will vary from mine!
- Grading can be ambiguous

Don’t let it rattle you!

Quick Quiz

Developers produce code. Who is the consumer of the code?

The Compiler!

- Your code HAS to meet the demands of the Compiler
- Compilers are NOT ambiguous!

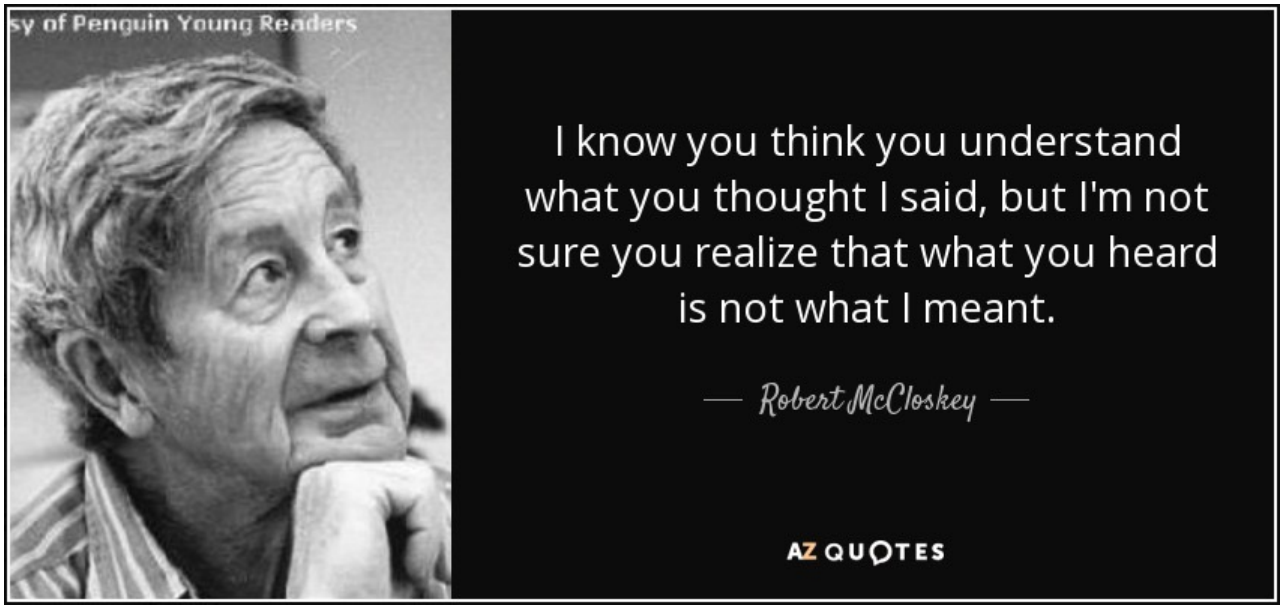
Did you say "end user"?

- Well, it's really nice if it satisfies the end user.
- Or at least it satisfies people who buy it (they may or may not be the end user.)

Architects produce architectures. Who is the consumer?

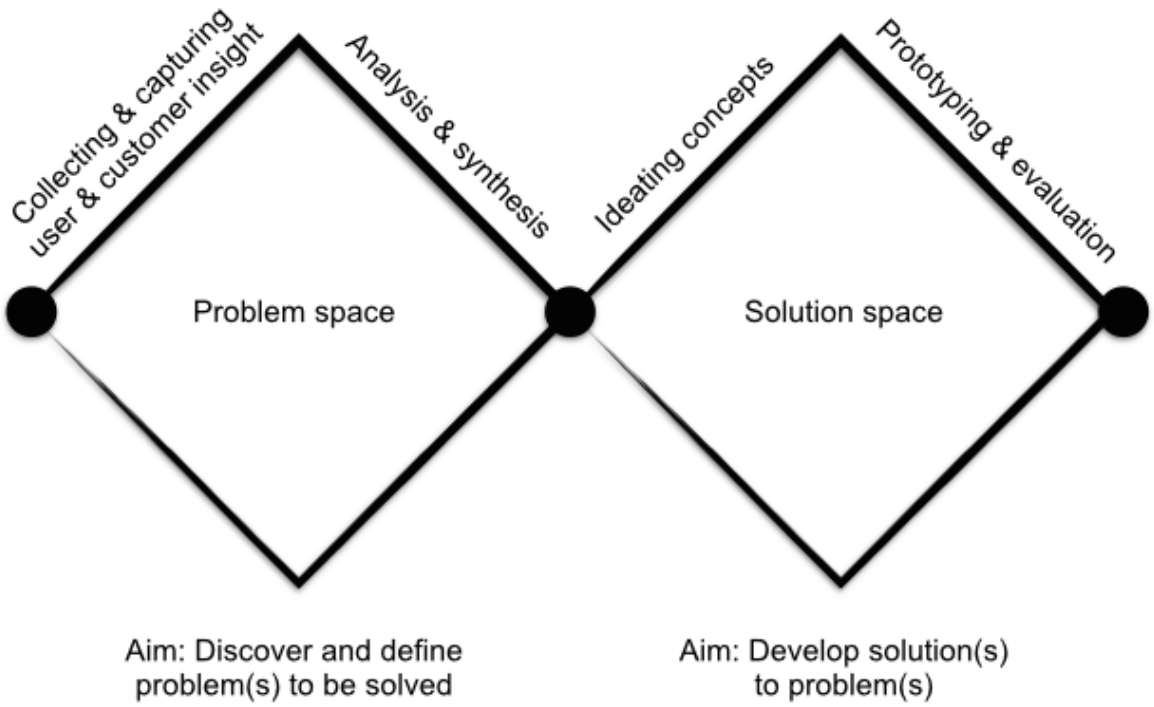
- PEOPLE! (including, but not limited to developers)
- And people are ambiguous!

Understanding is in the eye (and ear) of the beholder

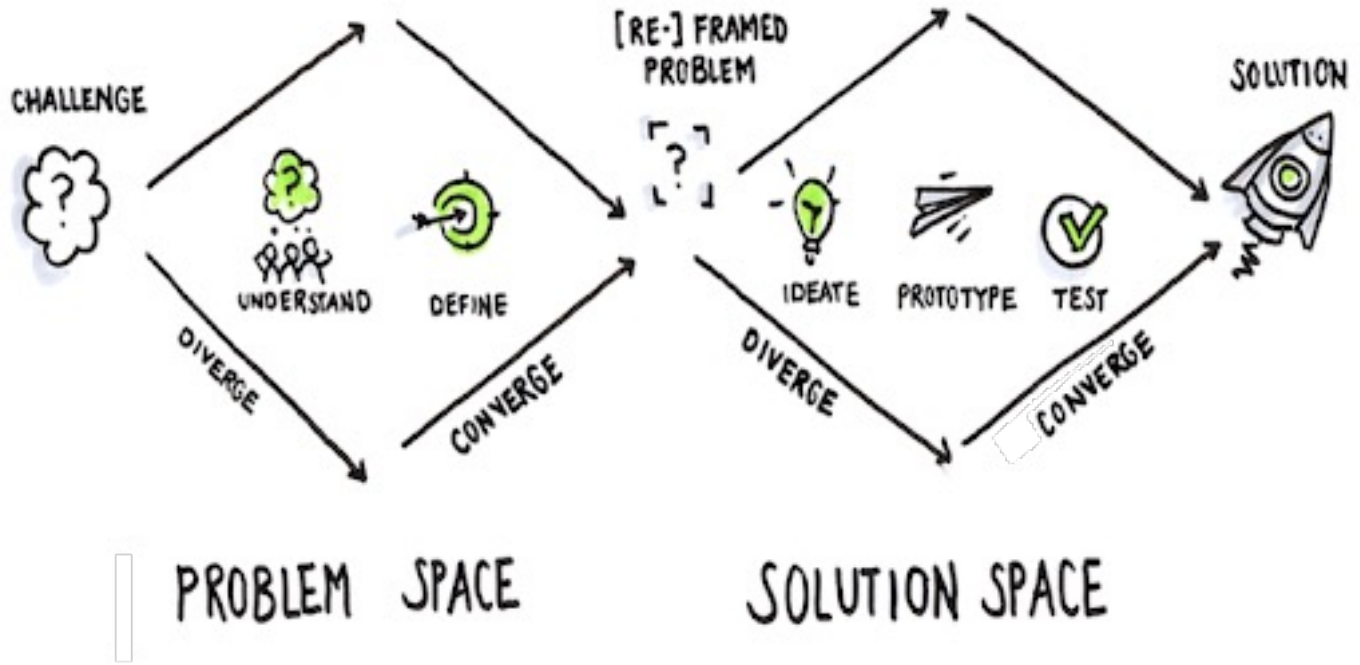


Problem Space Solution Space

Activities in each:



Place the architect in this diagram



Example #1

Problem Space vs. Solution Space

- Problem Space
  - A customer problem, need, or benefit that the product should address
  - A product requirement

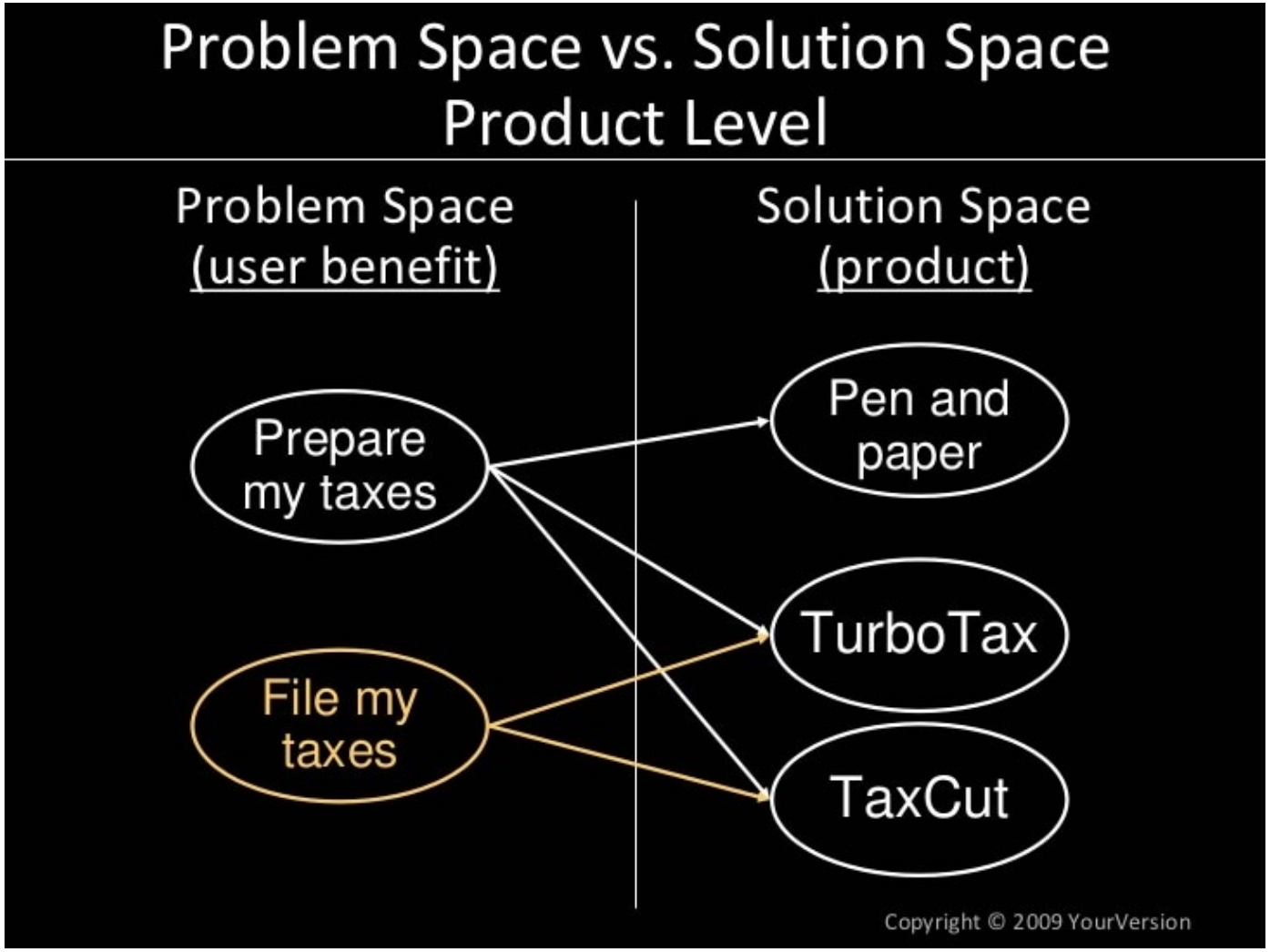
Example:

- Ability to write in space (zero gravity)

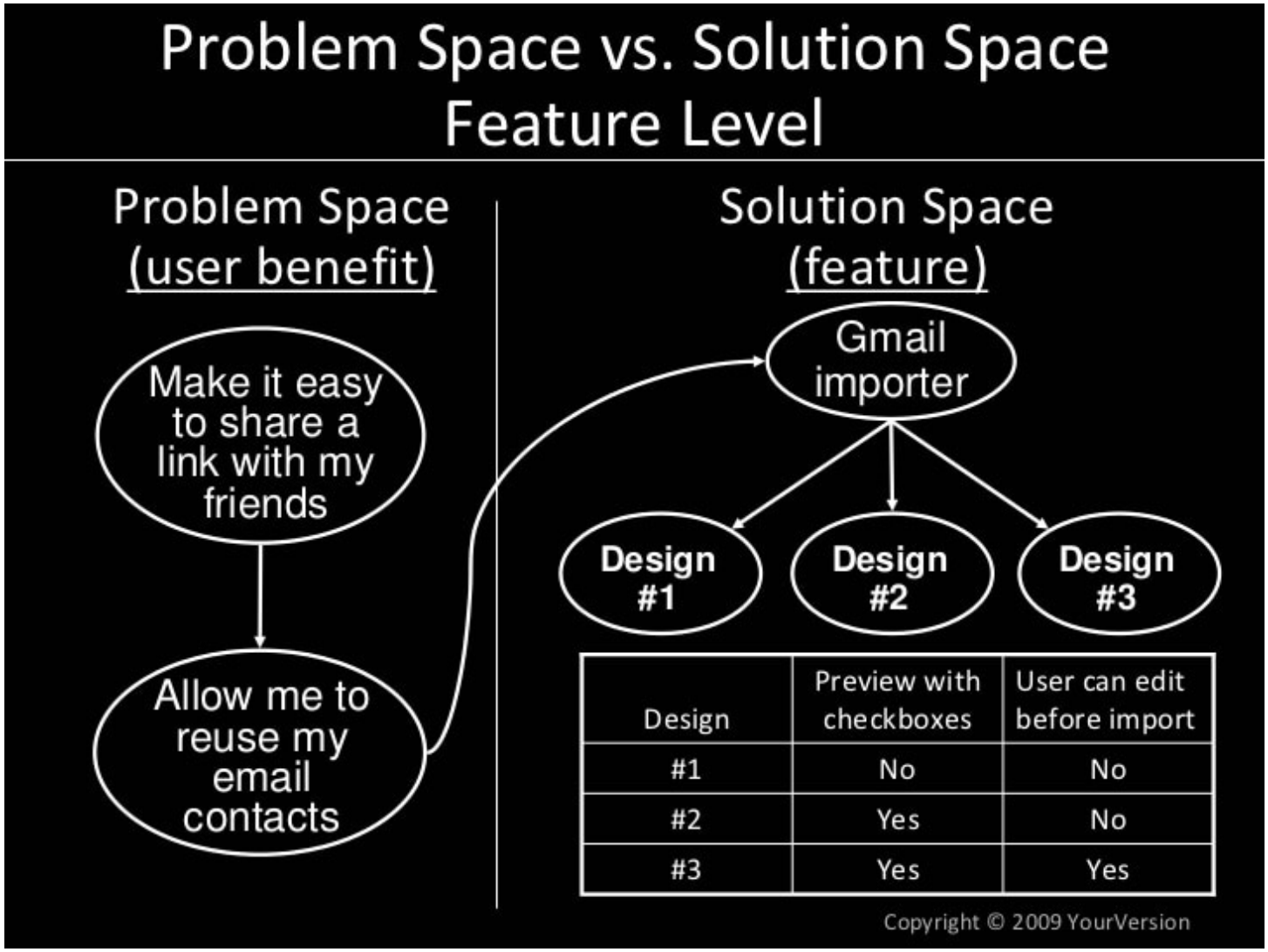
- Solution Space
  - A specific implementation to address the need or product requirement

- NASA: space pen ($1 M R&D cost)
- Russians: pencil

Example #2 (product level)



Example #3 (feature/architecture level)



A few words about roles

What role(s) usually work in the problem space?

Analysts, spec writers, surrogate users, Product Owners, etc.

And often architects!

What role(s) usually work in the solution space?

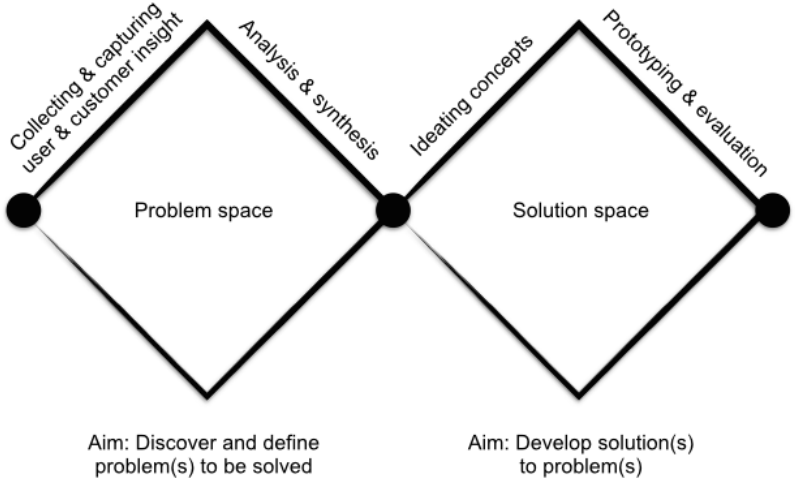
Architects, (other) designers, implementers, programmers, etc.

NOTE: in some companies, the architect does it all!

Not Waterfall

Early in the project:

Note that work in both diamonds proceeds ***simultaneously***.

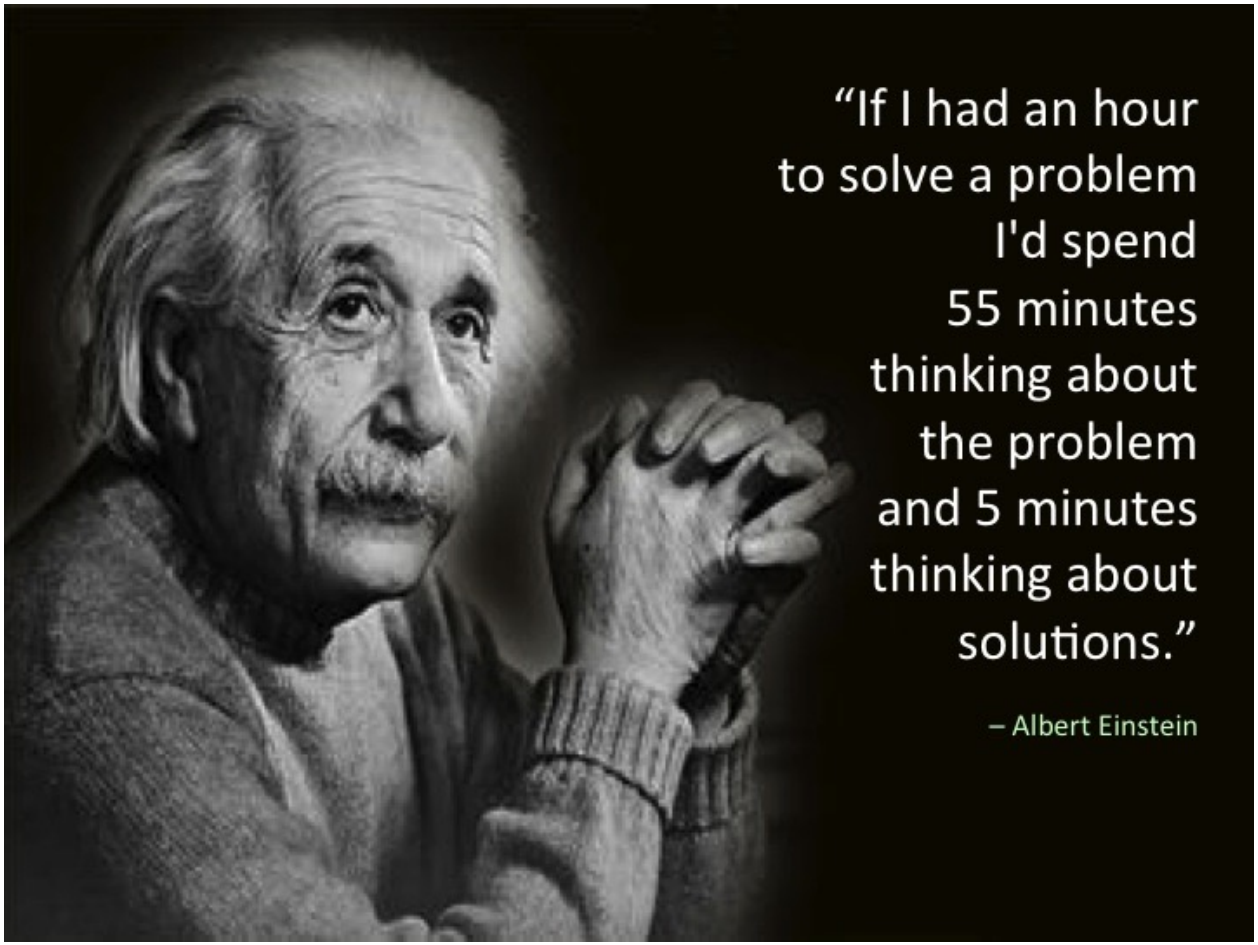


While you as architect are designing the architecture:

- The requirements are being analyzed

The things YOU learn are input to the problem space too!

- Can you think of examples?


“If I had an hour
to solve a problem
I'd spend
55 minutes
thinking about
the problem
and 5 minutes
thinking about
solutions.”
– Albert Einstein

Architecture Exercise

You are a lead developer in a company. The company makes software for medical billing: it manages billing of medical procedures first to insurance companies, and then bills remaining fees to customers. This is a very sophisticated system, as it must deal with multiple insurance companies (presumably each has its own API), as well as customers. And it should interface nicely with whatever software the doctor uses for the medical records and exam/treatment records.

One day your boss comes in and says, “I just got out of a meeting with the company executives. They see that there is a market opportunity for an app associated with our product. Our group will develop it. I’m appointing you to be the architect. Congratulations!”

What do you do?

1. Run screaming from the room
2. Accept, with some concerns
3. Call up your friend who is an app developer and say, “What’s an app?”
4. Start coding right away, and hope you can get a shell demo done by next week

Ok, you agreed to be the architect for the app. You no doubt have some questions.

Write the first question you will ask:

- (About the app itself, NOT about the schedule)

(only one question right now)

Attempt #1

Was your question: “Apple or Android”?

If so, 20 points from Slytherin!

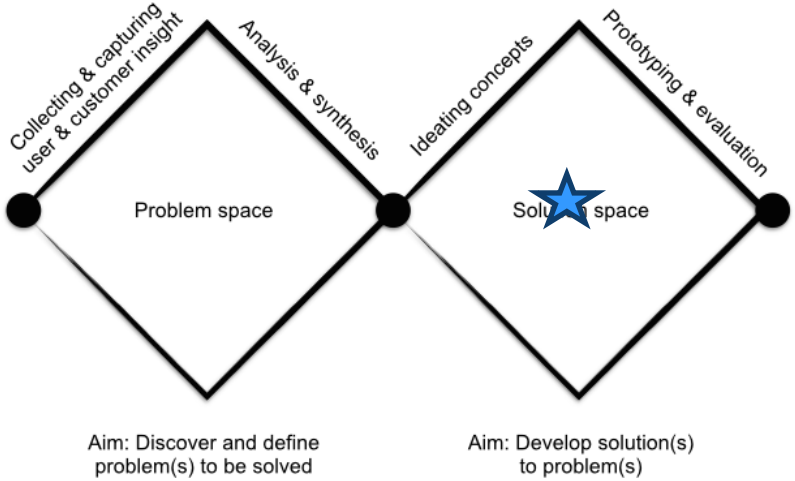
Why?

Because:

- This is way too early to ask this question.
- It IS an architectural question, but its focus is on implementation. Right now, you need more information on the other half: on the problem space rather than the solution space.
- So you have the wrong mindset.

Apple or Android?

Place this question in the diagram:



Attempt #2

Was your question: “Please give me a feature list for the App?”

If so, 10 points from Gryffindor!

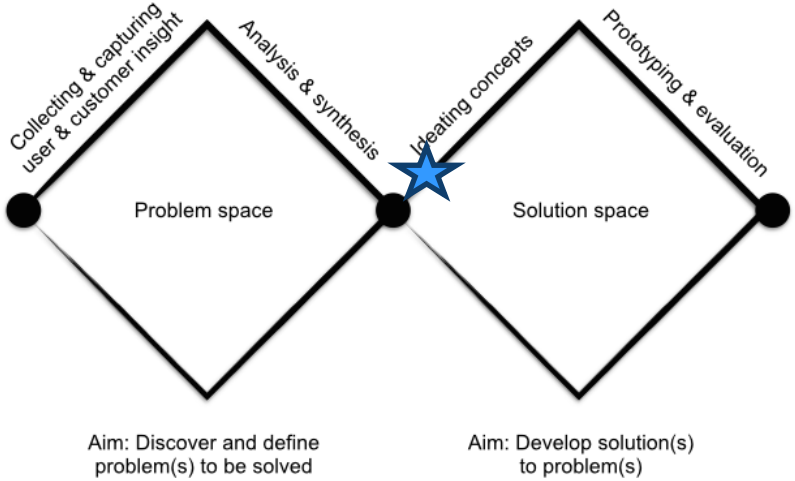
It’s pretty good. But what’s wrong with this question?

If you don’t know who will use it, you can’t know what it should do. Otherwise, it’s a solution looking for a problem.

This is also necessary for the architecture, but first things first!

What is the list of features?

Place this question in the diagram:



Attempt #3

Was your question: “Who will use the app?” or “Who is the app for?” (or similar)

Great! 20 points for Ravenclaw!

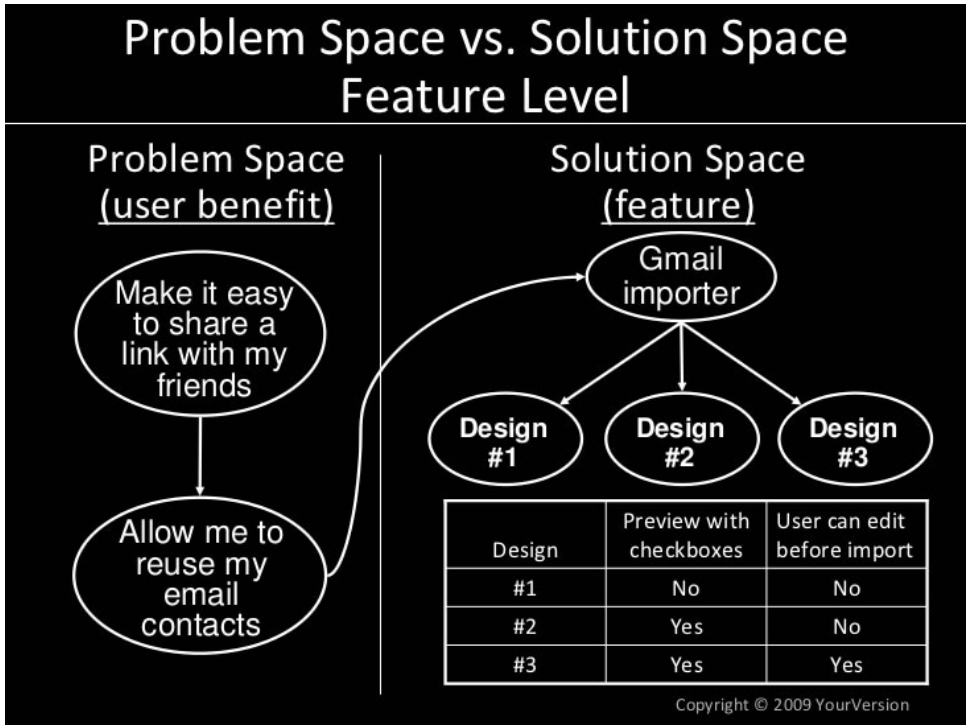
Why is this the starting point?

What similar questions will you want to also ask?

Ideas:

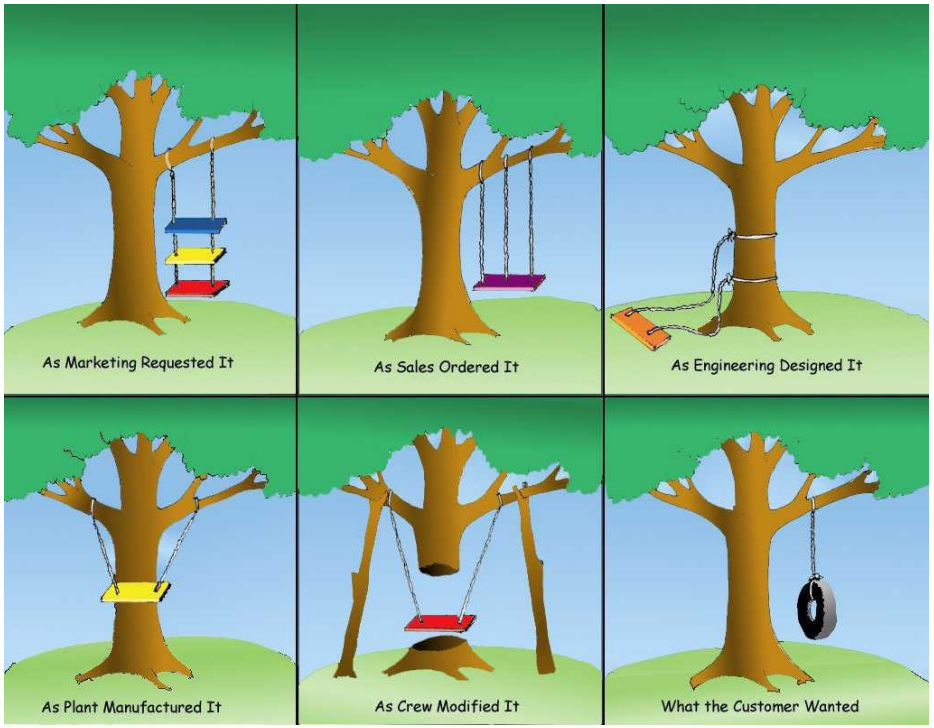
- What problem does the user have?
- What benefit will it be for the user of the app?
- Can they get that benefit anywhere else? (For example, suppose the app shows how much they currently owe. They might be able to get similar information from their insurance app. Or they can get through our company’s website, without using an app.)
- What is the compelling case for the app?

(Reminder: features are in the solution space)



Problem and Solution Space

As architects, we must understand the problem space



Analysis and Architecture

Analysis (of the problem space): figuring out what the user's problem is that we will solve in the product.

It's "requirements engineering"

That's not architecture!

But that looks like what you just did!

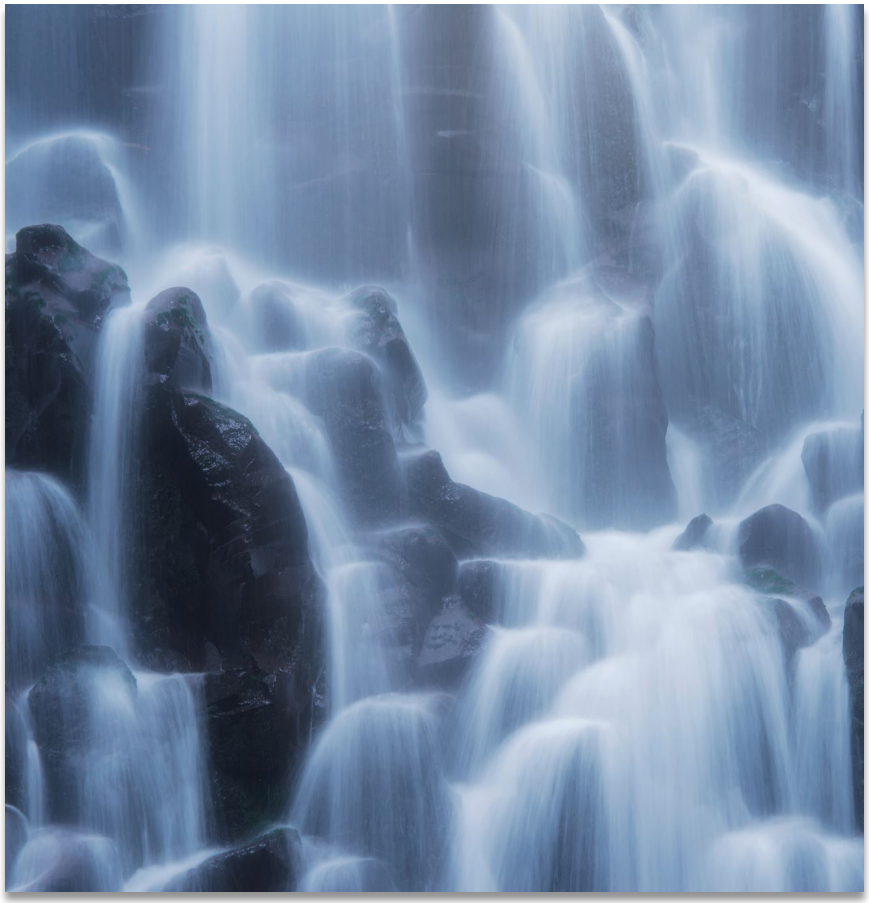
Since architecture is about a solution that solves the problem, architecture involves getting sufficient clarity of the problem space to create the solution

The Tip of the Waterfall

So why don't the analysts do their job, and get the requirements sufficiently specified?

Because they can't. At the very least, they need information from the architects

- Wait, isn't that backwards?
- No. (Think prototypes)



Analysts vs. Architects - another perspective

As an architect, you are asking questions about the nature of the problem, so you can understand the problem

- And then design a solution to the problem

Making sure you are solving the right problem

As an analyst, you dig up the answers to the questions

- Of course, you get to ask questions too!

Tangent (thought question):

- Where does marketing fit in this picture?

Just the Starting Point

You will ask LOTS more questions during architecture

Early on, they will be mostly in the problem space

Later, they will be more in the solution space

You will bounce between problem space and solution space

You want to learn a lot about the problem space: in particular, anything that will affect the architecture of the product.

Sometimes, “I don’t know” is the answer you will get.

Example: “Apple or Android” will come up, but just not the first thing

In-class exercise

Let's brainstorm all the questions an architect might ask about the proposed app

Here's the "spec" (the general idea) again:

An app for medical billing, related to the existing product: it manages billing of medical procedures first to insurance companies, and then bills remaining fees to customers. It interfaces with insurance companies, and with doctors' patient info software systems.

(note: some questions will depend on answers to other questions.)

The architect as a Bridge

But Remember!!!

The Architect MUST also know about the solution space:

You must know how to implement!!!

Reality!

If you design the architecture, can it actually be built?

Can you think of things in software that cannot be implemented?

Abstraction

"Applied Ambiguity"

Abstract Art

- What is this picture about?
- In 5 Minutes:
- Write down everything you can about this picture
- (Individual assignment)
- (artist: Pablo Picasso)

Abstract Art

What did you write down about the picture?

What were you able to infer from the picture, even though it is rather abstract?

Why?

From this experience, what is abstraction, and what can it do?

Abstraction

Abstraction is:

- The structured removal of information (you remove the details)
- Or: ignoring (not considering) details of something

OO Design

- We have base classes that are abstract
- It's more than that they can't be instantiated
- It's that they represent a family of related classes
- * And we can ***think*** about them in terms of the abstract class

But there are layers of abstraction above the OO Design

- Architecture

Architecture and Abstraction

Most architectural decisions are in the abstract area

We aren't worried about implementation details yet

Implementation *should* not impact the architecture

Problem: but sometimes implementation details do affect the architecture

- But often it's that we just didn't consider it back at architecture time

Goldilocks: not too little, not to much

A challenge in architecture:

- Too little abstraction (too much detail), and you get lost in the weeds
- Too much abstraction, and you don't have enough substance to work with.

The abstraction level changes with time

- Detail increases over time

But it's uneven: some things become detailed soon, others can be deferred until very late

Throughout the process, you always need to be able to find the abstract view of the system (the architecture; the theory)

Spikes of Detail

Occasionally, we have to go into considerable detail

- Because the outcome might affect the architecture
- Or because it's how to clarify requirements
- Or because we need to investigate feasibility
  - Example: feasibility of a new technology

Example:

- How fast does the automobile detection system need to be?
  - (What if there is heavy, but fast traffic?)
  - (Can our hardware keep up)
  - (What capacity computers will we need? Anything special?)
  - So we might do some traffic modeling and prototyping

Homework

It’s all about coming up with questions.

See homework description