# Introduct yourself

Hello, my name is Hoang

I have about 1 years experience as a golang developer.

My background is a back-end developer

Before I working with golang I used to work with asp.net core,

I have experience in front-end using reactjs

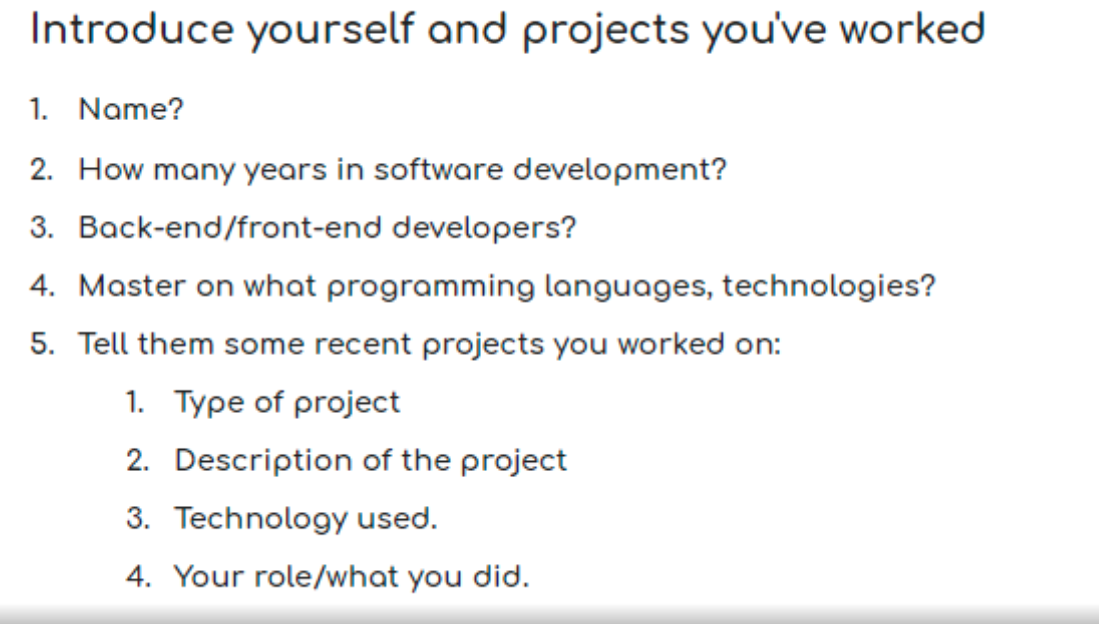
About databases, I have experience with nosql and sql databases, for example mongodb, postgresql, scrum

I have participate into 2 project:

First one is an e-commerces applications, the team size is around 10 people and my role was using ASP.NET core framework to create servies.

Second one is crawl data with golang. the team size is 7, my roles are use golang to create services.

technology use in project are Golang, mongodb, docker, postgresql.



Những câu hỏi chưa nghe bao giờ, hỏi lại nếu chưa nghe, nếu cần thì nhờ người ta ghi ra

Câu chưa biết, chưa có cơ hội làm việc với nó, nếu có cơ hội thì rất sẵn long

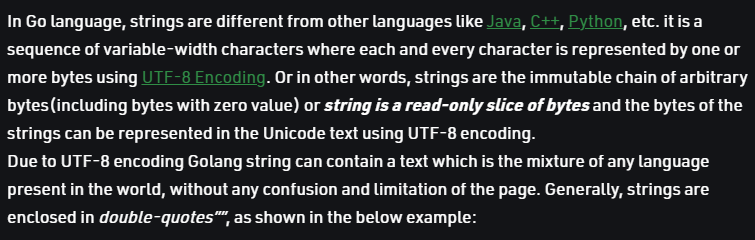
I haven't had a chance to work on it yet, if there is a chance I'd be happy to

Mình có thời gian để suy nghĩ

Deploy dự án, dùng docker, hoặc chạy trên server luôn, docker, docker compose

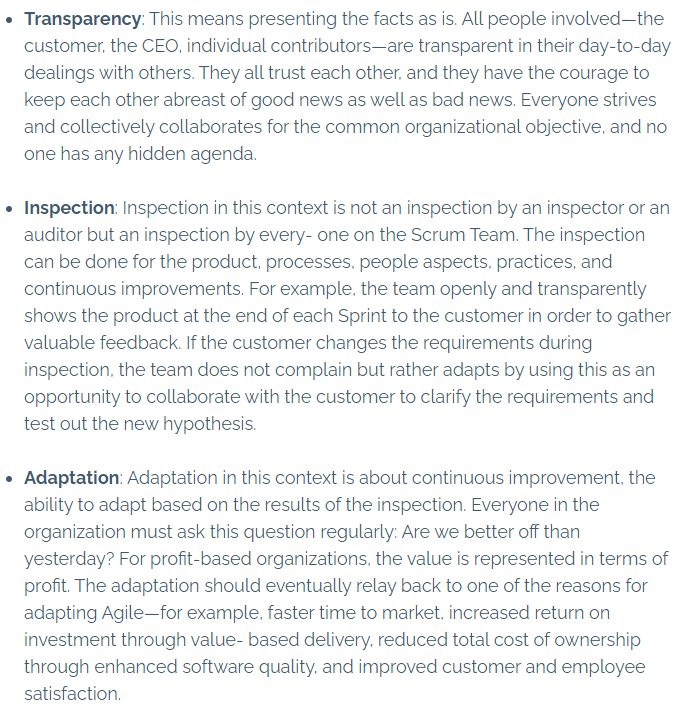
# String in golang, different with another language

<https://www.geeksforgeeks.org/strings-in-golang/>

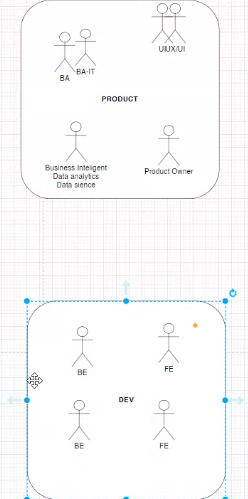
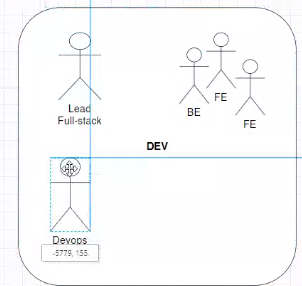
* 

# What is best scrum?

<https://www.scrum.org/resources/blog/three-pillars-empiricism-scrum>

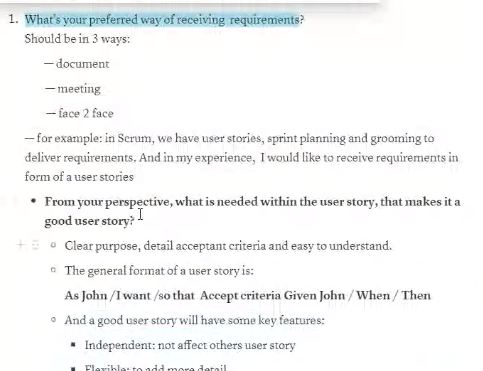


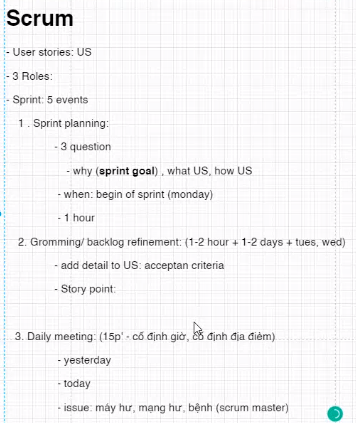




What’s your preferred way of receiving requirements?

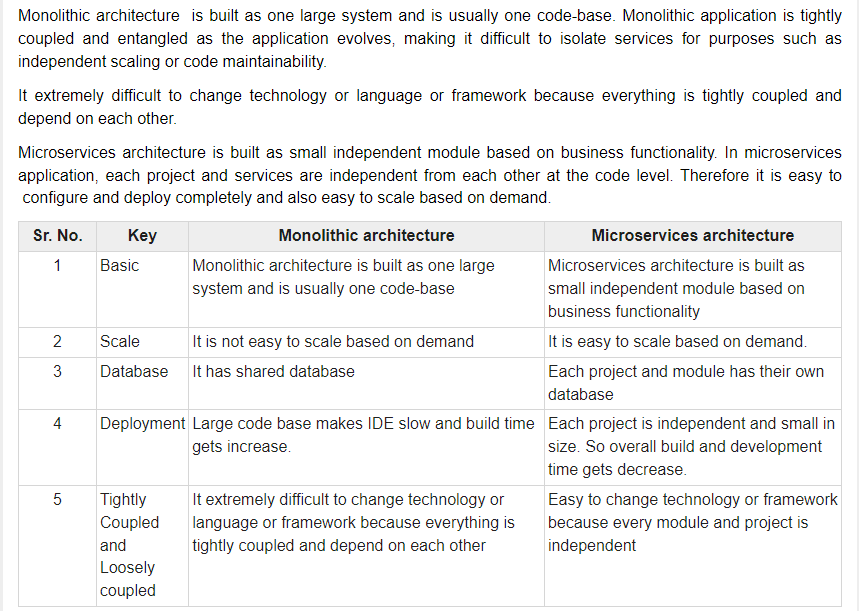




https://www.april9.com.au/blog/the-3-5-3-structure-of-scrum-what-it-is-why-it-works-and-why-you-should-be-using-it

# Microservices vs monolithic (architecture), different?

<https://www.tutorialspoint.com/difference-between-monolithic-and-microservices-architecture>

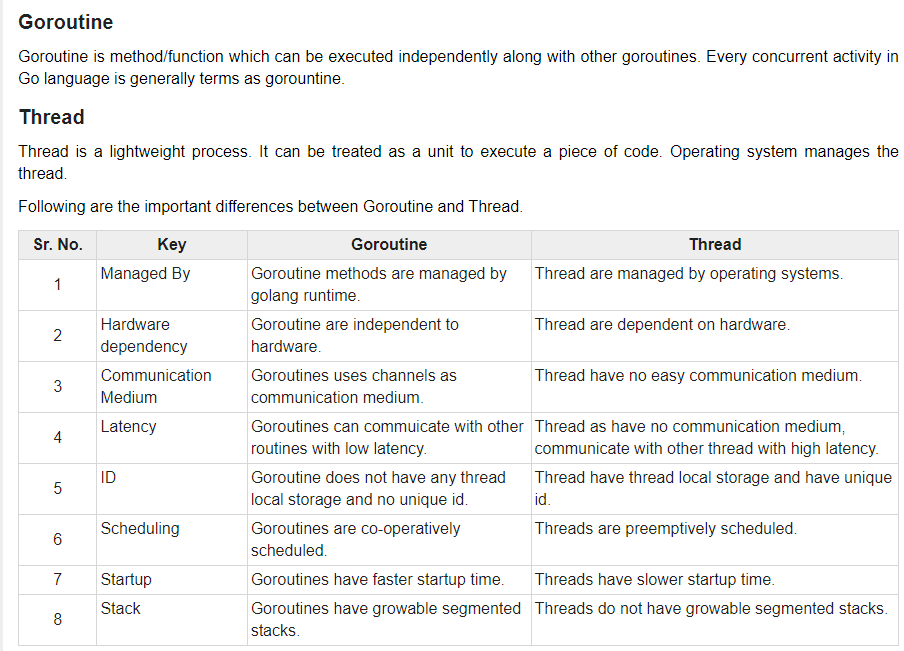


# What is docker?

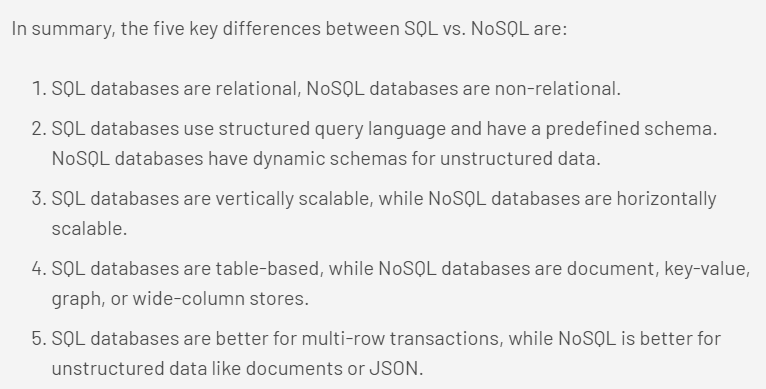
* Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. Because all of the containers share the services of a single operating system kernel, they use fewer resources than virtual machines.

# How to write a dockerfile – best practice

# Compare goroutine vs thread



# Different between SQL databases and noSQL databases

* 

# Dependency injection