Student’s name: Nguyễn Đăng Lộc

ID: SE160199

**1. What software development methodology would you suggest for this situation and why**

Answer:

Characteristics of given situation:

Requirements characteristics

* *Reliability:* the features desired and the activities of different kind of users on the system presented are not too detailed, so of the requirements are sill missing, incomplete, make this project is low-reliable-requirement one.
* *Types and number of requirements:* the brief of the context has already stated all the requirements of the system which contains both functional and non-functional requirements. The number of requirements is shown quite clear (not ambiguous, not vague), there are about more than 6 functional requirements and 2 non-functional requirements, showing that the project is not too big or too small, just a medium size and scope.
* *How often the requirements can change:* the business case is not too clear, the context has competely summaried some of the features required of only one role. This client haven’t had exactly thought of what they do with the system. Hence, the requirement is likely to change in the develop phase.
* *Can the requirements be defined at an early stage:* as mentioned earlier, the users’ need is not well identified, the problems is not clear enoug. The team implement the system may have to build, test and reconstruct the system. So, the requirement isn’t well-defined at an early stage, make this quite a adaptive project.

User involvement in the project:

- Large, because only some of the requirements of the clients’ role of the system is well-defined in advance, another requirements for another roles such as pet-sitter, managers are missing. Hence, to run this project, the developers need to communicate will business analyst/stakeholders/users to get feedback from them and build the right product in the final.

**In general:**

* **Vague Requirements** - the situation has just defined some features of the clients’ role, another roles’ requirements are not stated.
* **Unknown solution** – the client currently use Microsoft Excel as their solution, but when the demand increase, the old solution is no more suitable and adaptable. In the context, the client only stated that they want to have a new way to manage their business but not clearly what exactly they want, that’s mean they haven’t had any solution yet.
* **Need to deliver value fast and early** – since the old system using Microsoft Excel and is not adapted to increase demand of the market, the client need an innovation on their management system. The develop team have to work on implemention of new system such that has a quick speed to market. If the product is conducted slowly, the client may be lost their market share.
* **Client is available to work closely with development team** – In the situation has said that "The IT support of PetSitter could assist you in eliciate some missing requirements in the every stage."
* **High-risk level** – in the context given, there is no information about the resources, budgets, time, and another constraints may be involved in the project. If the leader of the project choosing a wrong working model, there could be a huge amount of waste on this project.

In conclusion, the user needs are not clear, the project need a high speed to market but the solution is not known fully and exactly, it’s very high-risk level. We need a iterative approach, some model may be fitted here are Unifed process, Increment and Iterative model, Agile mthod, etc. Morever, the client is avaible to collaborate with develop team, we can use **Scrum** for this project. This model will help develop team build each components, deliver in incremental approach, get feedback from client, then apply neccessary change for the next sprints.

**2. List out 4 functional requirements of clients’ role and 2 non-functional requirements of system.**

Functional requirements of clients’ role:

+ Clients can view schedule, request new services and request cancellation of existing services.

+ Clients can update his/her information and add pets.

+ Clients can view his/her Pet Care Journals and a gallery of all photos taken of his/her pets.

+ Clients can pay invoices.

Non-functional requirements of system:

+ Pet care provider use journals when clients want to view his/her pet care journals.

+ System has credit/debit payment functions that pet care provider can be enabled credit card processing when clients pay invoices.

**3. Two user stories of clients’ role.**

As a client, I want to request for new service or cancel an existing services so that I can freely choose my desired services of the pet care system.

As a client, I want to add my pets to the pet care system so that I can have my pet using the services of PetSitter.

**4. User story map for the clients’ role on the mobile app**

Table

Description automatically generated

**5.**

a)

I would not agree with the require that apply Waterfall Software Development to this project.

Waterfall method is a predictive method when all the requirements of the situation are identified clearly, and enough for the develop team to understand and implement. It’s used for repeated work that everything is known in advance, isn’t flexible for any change in the requirement. In constrast, the PetSitter system doesn’t have those characteristics

* The requirements are not well-defined, the develop team still have to learn more about another requirements to fulfill the activities of the system.
* Many features need to be explore of different kind of users on the system, the requirements of the whole system may be change in the future. Therefore, this is a adaptive approach and high-risk level project.
* The first release of Waterfall method takes a long time, while the client has a demand of quick apply the new management system to their business so as to adapt the considerably demand raising of the market and not lose market share.
* The high-risk level require developer to quickly find bug and fix it in right time. If we apply Waterfall, unluckily a bug is found in a latter phase of the develop cycle, it’s will be a terrible situation which cost a lot of resource and budget to refine.
* Last but not least, the context didn’t mention about the develop team experience in this domain, so apply a method which team experience may not be suitable in this case.

b)

The team should perform following testing stage (another additional detail stage if needed):

+ Component testing

+ Subsystem testing

+ Integration testing

+ System testing

+ Acceptance testing

Using a bottom up approach, the testing is expected to deliver the stub to test the individual components. After that, the team can build the driver to test if the components work well with its adjacent components and so on until the driver combines all components. The team perform subsytem testing to make sure the all the subsystem is working fine and present the right business value. Integration testing to identify and refine any problem when combine all the code together to be a whole system. And the last, I suggesst Verification & Validation process Testing to confirm that the software meet the user’s satisfaction and assure that the overall system operate as the user’s needs.

And the team must perform following type of testing:

1. Functional Testing - to confirm the correctness, efficiency, completeness of the functionality of the system.

2. Performance Testing - to make sure that the architecture of whole system and each components satisfies practical requirement in the real environment.

3. Scalability Testing - to make sure that the client get the benefits from the popularity growth of the product demand.

4. Usability Testing - to make sure that users are able to learn and use the system with no-effort, and in the user journey on the system there is no unexpected errors.

Testing team can check following for the quality of the system:

1. Do the system have all of functionality as stated in the requirements? Are they completed and efficient?

2. If the system sill operate stable when there is a huge amount user on the system?

3. Is there any defects in the functionalities?

4. Is the system able to scale up to meet the market growth demand?