# Agile:

3.

1. **Requirement characteristics:**

**Reliability:**

With all the reliability, this project are suitable for the Agile model because:

* Instead of relying solely on documentation, functioning software

demos are regarded as the greatest way to communicate with

consumers in the working software stage.

* Because the requirements cannot be fully acquired at the outset of the

project for a variety of reasons, customer collaboration is critical to

obtaining accurate product specifications.

* The final stage of Agile model development, Responding to change,

emphasises rapid response to change and continual improvement.

**Types and number of requirements:**.

* The most effective means of communication is face-to-face. Co- operation on a daily basis between business people and programmers..
* Any demand adjustments, no matter how last-minute, will be accommodated. Maintaining a focus on both technical prowess and aesthetic appeal. Adaptation to changing conditions on a regular basis.

**How often the requirements can change:**

* Agile models' flexibility allows for rapid shifts in needs. A realistic approach to software development, agile approaches encourage teamwork and cross-training, as well as rapid creation and demonstration of functionality, because it can handle both stagic and dynamic requirements,
* The agile methodology is able to produce early iterations of working solutions. A good model is chosen by the administration when dealing with situations that are both dynamic and manageable.

**Can the requirements be defined at an early stage**

* As a result of its limited rules and ease of use, the Agile model allows for continuous development and delivery within an overall planned context. Agile is superior to Waterfall or V-model because it gives developers more freedom.

1. **Development team:Team size:**

**Level of understanding of user requirements by the developers:**

* Due to the fact that Agile is a more team-based model than Waterfall, it is ideal for improving cross-training and interdepartmental cooperation and, as a result, fits the team Sell. Decisions that have a significant impact on the company's future are still made by the team lead.

1. **User involvement in the project: (Small/Average/Large):**
   * Throughout the development process, customers should be actively involved. It is their job to provide, priorities, and assess the new system needs and Iterations.
   * It's not uncommon for the project's requirements to shift on a regular basis. If a customer is eager to meet with a software development team at any time. the highly qualified and experienced Agile team is always available. This Agile project necessitates a small team and therefore a modest project.
2. **Conclusion:**

* In light of the foregoing evaluation, I feel that the Agile model is the most appropriate approach in this situation due to the frequent supply of requirements. Since the requirements system is flexible and subject to change at any time, it is appropriate to deploy the system first rather than developing it in stages.

# V-model:

1. **Requirement characteristics:**

* **Reliability:**

With all the reliability, this project are suitable for the V-model because:

* The Verification and Validation model (V-model) is the name given to this model. The V-Shaped life cycle follows a sequential course, much like the waterfall model. Before moving on to the next level, each one must be completed.
* The product will be tested concurrently with the development of a new version.
* Waterfall was the original inspiration for the V-model, a software development methodology. Requirements, specifications, design, and implementation are all part of the process, which includes a verification and validation step. This is done by unit testing, integration testing, system testing, and finally acceptance testing, which ensures that the requirements have been met.

* **Types and number of requirements:**
  + In the V-model, the project team identifies the high-level and detailed design phases of the project based on the project's needs. The requirements get more and more specific as each level is accomplished
* .**How often the requirements can change**
  + The V-shaped architecture is best suited for small to medium projects with well-defined and fixed requirements. It's best to go with a V- Shaped design if you have the necessary technological resources and expertise.
* **Can the requirements be defined at an early stage**.
  + Activities such as test planning and design take place long before code is written. This is a time-saver. As a result, the waterfall paradigm has a lower success rate.

1. **Development team:.**

**Team size:**

**Level of understanding of user requirements by the developers:**

* Unit tests, Integration tests, system tests, and acceptance tests are all included in the V-Model. The unit and integration tests ensure that the system's design is implemented in the code through the use of automated testing.
* Customer expectations are met through system and acceptability testing. Various components of the software are tested at various levels, each of which is performed independently of the others. To begin testing a new level, the V-model requires that the preceding level must be finished first.

1. **c. User involvement in the project: (Small/Average/Large)**
2. **Conclusion:**

* To summarize, I believe that the V-model is best suited to this situation, as the criteria are well defined and unlikely to alter. Because of the limited size of the group, each member should concentrate on a single stage at a time. Development in stages is unnecessary because the system's requirements don't specify which parts of the system should be deployed first.

# Waterfall

# Requirement characteristics:

**Reliability:**

**With all the reliability, this project are suitable for the Waterfall because:**

* Process Models were first introduced in the Waterfall Model. The term "linear- sequential file cycle model" is also used to describe this concept It's a breeze to learn and master.
* Using a waterfall paradigm, each stage must have been finished in full before the next stage can be started. Projects that are less than a year old and have no ambiguous requirements are ideal candidates for this paradigm of software development.
* It is only appropriate to apply this paradigm if the needs are well-known clearly defined, and unchanging.

**Types and number of requirements:**

* The business analyst gathers the requirements and the team analyzes them in this phase. During this stage, requirements are written down and more explanations can be obtained..
* The requirements are documented by the Business Analysts once they have spoken with the customer. The project team requires answers to the following questions, which were not included in the requirements paper, after going over and analyzing the requirements.

**How often the requirements can change:**.

* This is due to the stringent regulations and standards that must be adhered to.
* As a result, the requirements for these projects are well-known in advance, and contracts stipulate precisely what must be delivered when the project is completed:
* Design modifications can be implemented earlier in the Waterfall approach, which is advantageous because later design changes are difficult to implement Easy.
* In order to make this change, there is no need for any code or implementation.- Appropriate for projects with a tight deadline: The Waterfall model's sequential nature lends itself nicely to projects involving teams and organizations that thrive when given deadlines
* Team members will be able to readily grasp and follow the timeline if it is broken down into particular times. Furthermore, the researcher's job is made easier by having a timeline for the entire process and by establishing a few particular deadlines or milestones for each stage

**Can the requirements be defined at an early stage:**

* As a result of the model's rigidity, it's straightforward to keep track of the projects progress.
* Phases are done sequentially under this model. Do not mix phases.
* For smaller projects with well-defined and well-understood needs, the waterfall paradigm works well.

1. **Development team: Team size:**

**Level of understanding of user requirements by the developers:**

* The project's high- and low-level software architecture is the responsibility of the architect and other senior members of the team. To ensure that the banking application is always accessible, it has been decided that the system should have redundant backup and failover capabilities High-level and low-level design documents are drawn out by the architect.

1. **User involvement in the project: (Small/Average/Large)**
2. **Conclusion:**

* . To summarize. I believe that a waterfall approach would be preferable because the project's requirements are well defined and unlikely to change. Waterfall was also utilized in banking, healthcare, nuclear facilities, space shuttles and many other areas. Development in stages is unnecessary because the system's requirements don't specify which parts of the system should be deployed first.

Dịch :

# Nhanh nhẹn:

3.

1. **Đặc điểm yêu cầu:**

**Độ tin cậy:**

Với tất cả độ tin cậy, dự án này phù hợp với mô hình Agile vì:

* Thay vì chỉ dựa vào tài liệu, phần mềm hoạt động

Bản demo được coi là cách tốt nhất để giao tiếp với

người tiêu dùng trong giai đoạn phần mềm làm việc.

* Bởi vì các yêu cầu không thể có được đầy đủ ngay từ đầu

dự án vì nhiều lý do, sự hợp tác của khách hàng là rất quan trọng đối với

có được thông số kỹ thuật sản phẩm chính xác.

* Giai đoạn cuối cùng của phát triển mô hình Agile, Đáp ứng với sự thay đổi,

nhấn mạnh phản ứng nhanh với sự thay đổi và cải tiến liên tục.

**Các loại và số lượng yêu cầu:**.

* Phương tiện giao tiếp hiệu quả nhất là trực tiếp. Hợp tác hàng ngày giữa doanh nhân và lập trình viên.
* Bất kỳ điều chỉnh nhu cầu nào, bất kể vào phút cuối, sẽ được đáp ứng. Duy trì sự tập trung vào cả năng lực kỹ thuật và sự hấp dẫn thẩm mỹ. Thích ứng với các điều kiện thay đổi một cách thường xuyên.

**Tần suất các yêu cầu có thể thay đổi:**

* Tính linh hoạt của các mô hình Agile cho phép thay đổi nhu cầu nhanh chóng. Một cách tiếp cận thực tế để phát triển phần mềm, các phương pháp tiếp cận nhanh nhẹn khuyến khích làm việc theo nhóm và đào tạo chéo, cũng như tạo và trình diễn chức năng nhanh chóng, bởi vì nó có thể xử lý cả yêu cầu stagic và năng động,
* Phương pháp nhanh nhẹn có thể tạo ra các lần lặp lại sớm các giải pháp làm việc. Một mô hình tốt được chính quyền lựa chọn khi xử lý các tình huống vừa năng động vừa có thể quản lý được.

**Các yêu cầu có thể được xác định ở giai đoạn đầu**

* Do các quy tắc hạn chế và dễ sử dụng, mô hình Agile cho phép phát triển và phân phối liên tục trong bối cảnh được lên kế hoạch tổng thể. Agile vượt trội hơn so với Waterfall hoặc V-model vì nó mang lại cho các nhà phát triển sự tự do hơn.

1. **Nhóm phát triển: Quy mô nhóm:**

**Mức độ hiểu biết về yêu cầu của người dùng bởi các nhà phát triển:**

* Do thực tế là Agile là một mô hình dựa trên nhóm nhiều hơn Waterfall, nó lý tưởng để cải thiện đào tạo chéo và hợp tác giữa các bộ phận và do đó, phù hợp với nhóm Bán. Các quyết định có tác động đáng kể đến tương lai của công ty vẫn được đưa ra bởi trưởng nhóm.

1. **Sự tham gia của người dùng vào dự án: (Nhỏ / Trung bình / Lớn):**
   * Trong suốt quá trình phát triển, khách hàng nên tích cực tham gia. Công việc của họ là cung cấp, ưu tiên và đánh giá nhu cầu và lặp lại của hệ thống mới.
   * Không có gì lạ khi các yêu cầu của dự án thay đổi một cách thường xuyên. Nếu khách hàng háo hức gặp gỡ nhóm phát triển phần mềm bất cứ lúc nào. đội ngũ Agile có trình độ cao và giàu kinh nghiệm luôn sẵn sàng. Dự án Agile này đòi hỏi một nhóm nhỏ và do đó là một dự án khiêm tốn.
2. **Kết thúc:**

* Dựa trên đánh giá ở trên, tôi cảm thấy rằng mô hình Agile là cách tiếp cận phù hợp nhất trong tình huống này do việc cung cấp các yêu cầu thường xuyên. Vì hệ thống yêu cầu linh hoạt và có thể thay đổi bất cứ lúc nào, nên việc triển khai hệ thống trước là phù hợp hơn là phát triển nó theo từng giai đoạn.

# Mô hình chữ V:

1. **Đặc điểm yêu cầu:**

* **Độ tin cậy:**

Với tất cả độ tin cậy, dự án này phù hợp với mô hình V vì:

* Mô hình Xác minh và Xác thực (mô hình V) là tên được đặt cho mô hình này. Vòng đời hình chữ V tuân theo một lộ trình tuần tự, giống như mô hình thác nước. Trước khi chuyển sang cấp độ tiếp theo, mỗi cấp độ phải được hoàn thành.
* Sản phẩm sẽ được thử nghiệm đồng thời với sự phát triển của một phiên bản mới.
* Thác nước là nguồn cảm hứng ban đầu cho mô hình V, một phương pháp phát triển phần mềm. Các yêu cầu, thông số kỹ thuật, thiết kế và triển khai đều là một phần của quy trình, bao gồm bước xác minh và xác nhận. Điều này được thực hiện bằng cách kiểm tra đơn vị, kiểm tra tích hợp, kiểm tra hệ thống và cuối cùng là kiểm tra chấp nhận, đảm bảo rằng các yêu cầu đã được đáp ứng.

* **Các loại và số lượng yêu cầu:**
  + Trong mô hình V, nhóm dự án xác định các giai đoạn thiết kế cấp cao và chi tiết của dự án dựa trên nhu cầu của dự án. Các yêu cầu ngày càng cụ thể hơn khi mỗi cấp độ được hoàn thành
* . **Tần suất các yêu cầu có thể thay đổi**
  + Kiến trúc hình chữ V phù hợp nhất cho các dự án vừa và nhỏ với các yêu cầu được xác định rõ ràng và cố định. Tốt nhất bạn nên sử dụng thiết kế hình chữ V nếu bạn có đủ nguồn lực và chuyên môn công nghệ cần thiết.
* **Các yêu cầu có thể được xác định ở giai đoạn đầu**.
  + Các hoạt động như lập kế hoạch và thiết kế bài kiểm tra diễn ra rất lâu trước khi mã được viết. Đây là một tiết kiệm thời gian. Do đó, mô hình thác nước có tỷ lệ thành công thấp hơn.

1. **Nhóm phát triển:.**

**Quy mô nhóm:**

**Mức độ hiểu biết về yêu cầu của người dùng bởi các nhà phát triển:**

* Kiểm tra đơn vị, Kiểm tra tích hợp, Kiểm tra hệ thống và Kiểm tra chấp nhận đều được bao gồm trong Mô hình V. Các bài kiểm tra đơn vị và tích hợp đảm bảo rằng thiết kế của hệ thống được thực hiện trong mã thông qua việc sử dụng kiểm tra tự động.
* Kỳ vọng của khách hàng được đáp ứng thông qua hệ thống và kiểm tra khả năng chấp nhận. Các thành phần khác nhau của phần mềm được kiểm tra ở nhiều cấp độ khác nhau, mỗi cấp độ được thực hiện độc lập với các thành phần khác. Để bắt đầu thử nghiệm một cấp độ mới, mô hình V yêu cầu rằng cấp độ trước đó phải được hoàn thành trước.

1. **c. Sự tham gia của người dùng vào dự án: (Nhỏ/Trung bình/Lớn)**
2. **Kết thúc:**

* Tóm lại, tôi tin rằng mô hình V phù hợp nhất với tình huống này, vì các tiêu chí được xác định rõ và không có khả năng thay đổi. Do quy mô của nhóm có hạn, mỗi thành viên nên tập trung vào một giai đoạn duy nhất tại một thời điểm. Việc phát triển theo từng giai đoạn là không cần thiết vì các yêu cầu của hệ thống không nêu rõ phần nào của hệ thống nên được triển khai trước.

# Thác nước

# Đặc điểm yêu cầu:

**Độ tin cậy:**

**Với tất cả độ tin cậy, dự án này phù hợp với Thác nước vì:**

* Mô hình quy trình lần đầu tiên được giới thiệu trong Mô hình Thác nước. Thuật ngữ "mô hình chu kỳ tệp tuyến tính- tuần tự" cũng được sử dụng để mô tả khái niệm này Thật dễ dàng để học và thành thạo.
* Sử dụng mô hình thác nước, mỗi giai đoạn phải được hoàn thành đầy đủ trước khi giai đoạn tiếp theo có thể được bắt đầu. Các dự án chưa đầy một năm tuổi và không có yêu cầu mơ hồ là ứng cử viên lý tưởng cho mô hình phát triển phần mềm này.
* Chỉ thích hợp để áp dụng mô hình này nếu các nhu cầu được xác định rõ ràng và không thay đổi.

**Các loại và số lượng yêu cầu:**

* Nhà phân tích kinh doanh thu thập các yêu cầu và nhóm phân tích chúng trong giai đoạn này. Trong giai đoạn này, các yêu cầu được viết ra và có thể thu được nhiều giải thích hơn.
* Các yêu cầu được ghi lại bởi các Nhà phân tích kinh doanh sau khi họ đã nói chuyện với khách hàng. Nhóm dự án yêu cầu câu trả lời cho các câu hỏi sau, không có trong giấy yêu cầu, sau khi xem xét và phân tích các yêu cầu.

**Tần suất các yêu cầu có thể thay đổi:**.

* Điều này là do các quy định và tiêu chuẩn nghiêm ngặt phải được tuân thủ.
* Do đó, các yêu cầu đối với các dự án này đã được biết trước và các hợp đồng quy định chính xác những gì phải được giao khi dự án hoàn thành:
* Sửa đổi thiết kế có thể được thực hiện sớm hơn trong cách tiếp cận Thác nước, điều này có lợi vì những thay đổi thiết kế sau này rất khó thực hiện Dễ dàng.
* Để thực hiện thay đổi này, không cần bất kỳ mã hoặc triển khai nào.- Thích hợp cho các dự án có thời hạn chặt chẽ: Bản chất tuần tự của mô hình Thác nước phù hợp với các dự án liên quan đến các nhóm và tổ chức phát triển mạnh khi có thời hạn nhất định
* Các thành viên trong nhóm sẽ có thể dễ dàng nắm bắt và theo dõi dòng thời gian nếu nó được chia thành các thời điểm cụ thể. Hơn nữa, công việc của nhà nghiên cứu được thực hiện dễ dàng hơn bằng cách có một mốc thời gian cho toàn bộ quá trình và bằng cách thiết lập một vài thời hạn hoặc cột mốc cụ thể cho từng giai đoạn

**Các yêu cầu có thể được xác định ở giai đoạn đầu:**

* Do độ cứng của mô hình, thật đơn giản để theo dõi tiến độ dự án.
* Các giai đoạn được thực hiện tuần tự theo mô hình này. Không trộn lẫn các pha.
* Đối với các dự án nhỏ hơn với nhu cầu được xác định rõ ràng và được hiểu rõ, mô hình thác nước hoạt động tốt.

1. **Nhóm phát triển: Quy mô nhóm:**

**Mức độ hiểu biết về yêu cầu của người dùng bởi các nhà phát triển:**

* Kiến trúc phần mềm cấp cao và cấp thấp của dự án là trách nhiệm của kiến trúc sư và các thành viên cấp cao khác trong nhóm. Để đảm bảo rằng ứng dụng ngân hàng luôn có thể truy cập được, người ta đã quyết định rằng hệ thống nên có khả năng sao lưu dự phòng và chuyển đổi dự phòng Các tài liệu thiết kế cấp cao và cấp thấp được kiến trúc sư rút ra.

1. **Sự tham gia của người dùng vào dự án: (Nhỏ / Trung bình / Lớn)**
2. **Kết thúc:**

* . Tóm lại. Tôi tin rằng cách tiếp cận thác nước sẽ thích hợp hơn vì các yêu cầu của dự án được xác định rõ ràng và không có khả năng thay đổi. Thác nước cũng được sử dụng trong ngân hàng, chăm sóc sức khỏe, cơ sở hạt nhân, tàu con thoi và nhiều lĩnh vực khác. Việc phát triển theo từng giai đoạn là không cần thiết vì các yêu cầu của hệ thống không nêu rõ phần nào của hệ thống nên được triển khai trước.

Dịch :