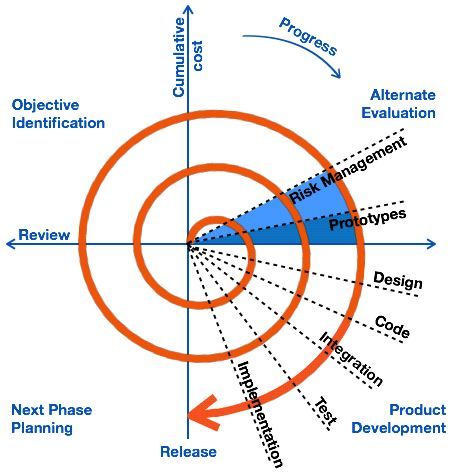
1. *Nêu ý tưởng, các pha, ưu điểm và nhược điểm và áp dụng mô hình của mô hình Xoắn Ốc*
2. *Nêu ý tưởng, các pha, ưu điểm và nhược điểm mô hình của mô hình chữ V*
3. *So sánh hai mô hình trên*

**Mô hình xoắn ốc**



**Ý tưởng:**

* Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
* Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
* Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, …

**Phân tích mô hình**

Các pha trong quy trình phát triển xoắn ốc bao gồm:

* **Objective identification- Thiết lập mục tiêu**: xác định mục tiêu, đối tượng cho từng pha của dự án.
* **Alternate evaluation- Đánh giá và giảm thiểu rủi ro**: đánh giá rủi ro và thực hiện các hành động để giảm thiểu rủi ro.
* **Product development- Phát triển sản phẩm**: Lựa chọn mô hình phù hợp để phát triển hệ thống.
* **Next phase planning- Lập kế hoạch**: đánh giá dự án và lập kế hoạch cho pha tiếp theo.

**Áp dụng:**

Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn.

**Ưu điểm**

* Tốt cho các hệ phần mềm quy mô lớn.
* Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa.
* Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.

**Nhược điểm**

* Manager cần có kỹ năng tốt để quản lý dự án, đánh giá rủi ro kịp thời.
* Chi phí cao và mất nhiều thời gian để hoàn thành dự án.
* Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro.
* Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn.
* Chưa được dùng rộng rãi.

**Mô hình chữ V( V model)**

# **I. Mô hình phát triển phần mềm V là gì?**

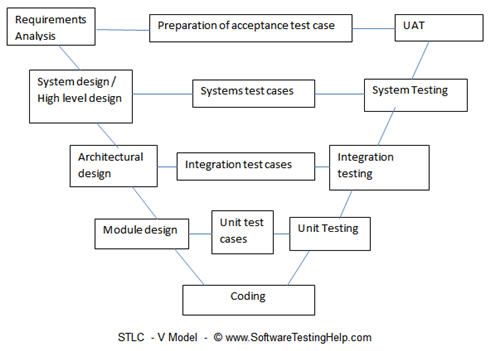
Mô hình V hiện nay là một trong những quy trình phát triển phần mềm được sử dụng rộng rãi nhất. Trong mô hình V việc thực hiện kiểm tra được diễn ra ngay từ giai đoạn lấy yêu cầu. V mô hình cũng được gọi là mô hình xác minh (verification) và mô hình xác nhận (validation).

Để hiểu được mô hình V, trước hết chúng ta hãy hiểu xác minh (verification) và xác nhận hợp lệ (validation) trong phần mềm là gì.

**Xác minh (verification) :** Xác minh là một kỹ thuật phân tích tĩnh. Trong kiểm thử, kỹ thuật này được thực hiện mà không phải chạy code. Nó bao gồm một số hoạt đông như xem lại (review), kiểm tra (inspection) và kiểm tra từ đầu tới cuối (walkthrough).

**Xác nhận (validation):** Xác nhận là một kỹ thuật phân tích động, trong đó việc kiểm thử được thực hiện bằng cách thực hiện code. Ví dụ bao gồm kỹ thuật kiểm tra chức năng (function) và phi chức năng (non-function).

Trong mô hình V, các hoạt động phát triển và đảm bảo chất lượng được thực hiện đồng thời. Không có pha rời rạc được gọi là kiểm thử, thay vào đó kiểm thử được bắt đầu ngay từ giai đoạn lấy yêu cầu. Các hoạt động xác minh và xác nhận đi liền với nhau. Để hiểu mô hình V, chúng ta hãy nhìn vào hình dưới đây:



Trong một quá trình phát triển theo mô hình V điển hình, bên trái là các hoạt động phát triển và bên tay phải là các hoạt động kiểm thử. Trong giai đoạn phát triển cả việc xác minh và xác nhận được thực hiện cùng với các hoạt động phát triển thực tế. Bây giờ chúng ta hãy tìm hiểu bức tranh phía trên rõ hơn:

## Phía bên tay trái:

Như đã nói ở trên, hoạt động phía bên trái là hoạt động phát triển. Thông thường, chúng ta khó có thể thực hiện thử nghiệm trong giai đoạn phát triển, nhưng đây là ưu điểm của mô hình này, nó chứng tỏ rằng thử nghiệm cũng có thể được thực hiện ở tất cả các giai đoạn của các hoạt động phát triển.

### 1. Phân tích yêu cầu:

Trong giai đoạn này các yêu cầu được thu thập, phân tích và nghiên cứu. Ở giai đoạn này việc hệ thống chạy như thế nào không quan trọng, quan trọng là hệ thống có những chức năng gì. Brain storming/walkthrough, interviews cần được thực hiện để có mục tiêu rõ ràng.

* Hoạt động xác minh:Đánh giá yêu cầu (Requirements review).
* Hoạt động xác nhận: Tạo test case UAT (User acceptance test- kiểm thử chấp nhận) = Đầu ra cần có: Tài liệu hiểu về yêu cầu, UAT test case.

### 2. Yêu cầu hệ thống / High level design:

Trong giai đoạn này, high level design của phần mềm được xây dựng. Nhóm sẽ nghiên cứu và điều tra về các yêu cầu có thể được thực hiện như thế nào. Tính khả thi về mặt kỹ thuật của yêu cầu cũng được tìm hiểu. Nhóm cũng tìm hiểu về các mô-đun sẽ được tạo / phụ thuộc, nhu cầu phần cứng / phần mềm

* Hoạt động xác minh: Đánh giá thiết kế (Design reviews)
* Hoạt động xác nhận: Tạo test plan và test case, tạo ma trân truy vết (traceability metrics)
* Đầu ra cần có: System test cases, Feasibility reports, System test plan, tài liệu về yêu cầu phần cứng và các mô đun, ...

### 3. Thiết kế kiến trúc:

Trong giai đoạn này, dựa trên thiết kế mức cao, kiến trúc phần mềm được tạo ra. Các mô-đun, mối quan hệ và sự phụ thuộc của họ, sơ đồ kiến trúc, bảng cơ sở dữ liệu, chi tiết về công nghệ đều được hoàn tất trong giai đoạn này.

* Hoạt động xác minh: Đánh giá thiết kế
* Hoạt động xác nhận: Kế hoạch thử nghiệm tích hợp và các trường hợp thử nghiệm.
* Đầu ra cần có: Tài liệu thiết kế, Kế hoạch kiểm thử tích hợp và các trường hợp thử nghiệm, Thiết kế bảng cơ sở dữ liệu,...

### 4. Thiết kế mô đun / Thiết kế cấp thấp:

Trong giai đoạn này mỗi mô-đun hoặc các thành phần phần mềm đều được thiết kế riêng. Các method, class, giao diện, các kiểu dữ liệu vv đều được hoàn tất trong giai đoạn này.

* Hoạt động xác minh: Đánh giá thiết kế
* Hoạt động xác nhận: Tạo và xem xét các trường hợp kiểm tra đơn vị.
* Đầu ra cần có: Các đơn vị kiểm tra đơn vị

### 5. Code:

Trong giai đoạn này, code được thực hiện.

* Hoạt động xác minh: Xem xét mã, kiểm tra các trường hợp kiểm tra
* Hoạt động xác nhận: Tạo các trường hợp kiểm tra chức năng.
* Đầu ra cần có: các trường hợp thử nghiệm, danh sách kiểm tra xem lại.

## Bên tay phải:

Phía bên phải thể hiện các hoạt động kiểm thử hoặc Giai đoạn Xác nhận. Chúng ta sẽ bắt đầu đi từ phía dưới.

### 1. Kiểm tra đơn vị (unit test):

Trong giai đoạn này tất cả unit test case đã được tạo ra trong giai đoạn thiết kế cấp thấp sẽ được thực hiện.

* Unit test là một kỹ thuật kiểm tra hộp trắng, nơi một đoạn code được viết, nó sẽ tích hợp một phương pháp (hoặc một đoạn code khác) để kiểm tra xem đoạn code có cho kết quả mong muốn hay không. Thử nghiệm này về cơ bản được thực hiện bởi nhóm phát triển. Trong trường hợp có bất thường, bug được ghi lại và theo dõi.
* Đầu ra cần có: Kết quả thực hiện unit test

### 2. Kiểm tra tích hợp:

Trong giai đoạn này các trường hợp kiểm thử tích hợp được thực hiện đã được tạo ra trong giai đoạn thiết kế kiến trúc. Trong trường hợp có bất kỳ dị thường, bug được ghi lại và theo dõi.

* Kiểm thử tích hợp (Integration Testing): Kiểm thử tích hợp là một kỹ thuật mà đơn vị kiểm tra là mô-đun được tích hợp và kiểm tra xem các mô-đun tích hợp vào có cho kết quả mong đợi không. Nói một cách đơn giản, nó xác nhận xem các thành phần của ứng dụng có hoạt động với nhau như mong đợi hay không.
* Đầu ra cần có: Các kết quả kiểm tra tích hợp.

### 3. Kiểm tra hệ thống (System testing):

Trong giai đoạn này, kiểm thử chức năng và phi chức năng được thực hiện. Nói cách khác, việc kiểm tra thực tế hoạt động của ứng dụng diễn ra ở đây. Lỗi được phát hiện và theo dõi để sửa. Báo cáo tiến độ cũng là một phần quan trọng trong giai đoạn này. Ma trận truy vết (traceability metrics) được cập nhật để kiểm tra mức độ bao phủ và rủi ro được giảm bớt.

* Đầu ra cần có: Kết quả kiểm tra, Các bản ghi kiểm tra, báo cáo lỗi, báo cáo tóm tắt kiểm tra và các ma trận truy xuất cập nhật.

### 4. Thử nghiệm chấp nhận của người dùng:

Thử nghiệm chấp nhận về cơ bản liên quan đến việc kiểm tra các yêu cầu business. Ở đây kiểm tra được thực hiện để xác nhận rằng các yêu cầu kinh doanh được đáp ứng trong môi trường người dùng. Thử nghiệm khả năng tương thích và đôi khi thử nghiệm phi chức năng (Load, Stress và Volume) cũng được thực hiện trong giai đoạn này.

* Đầu ra cần có: Kết quả UAT, Ma trận độ bao phủ business được cập nhật.

# **II. Khi nào sử dụng V Model?**

V được áp dụng khi:

* Yêu cầu được xác định rõ ràng và không mơ hồ
* Tiêu chí chấp nhận được xác định rõ ràng.
* Dự án có quy mô vừa và nhỏ.
* Công nghệ và công cụ được sử dụng không thường xuyên thay đổi.

# **III. Ưu điểm và nhựơc điểm khi sử dụng mô hình V:**

## 1. Ưu điểm:

* Quá trình phát triển và quy trình quản lý có tính tổ chức và hệ thống
* Hoạt động tốt cho các dự án có quy mô vừa và nhỏ.
* Kiểm tra bắt đầu từ khi bắt đầu phát triển vì vậy sự mơ hồ được xác định ngay từ đầu.
* Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

## 2. Nhược điểm:

* Không thích hợp cho các dự án lớn và phức tạp
* Không phù hợp nếu các yêu cầu thường xuyên thay đổi.
* Không có phần mềm làm việc được sản xuất ở giai đoạn trung gian.
* Không có điều khoản cho việc phân tích rủi ro nên có sự không chắc chắn và có tính rủi ro.

# SO SÁNH MÔ HÌNH CHỮ V VỚI MÔ HÌNH XOẮN ỐC:

|  |  |
| --- | --- |
| Mô hình Xoắn Ốc | Mô hình chữ V |
| **Ý tưởng:**   * Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước. * Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp. * Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, … | **Ý tưởng:**   * Mô hình chữ V là một phần mở rộng của mô hình thác nước và được dựa trên sự kết hợp của một giai đoạn thử nghiệm cho từng giai đoạn phát triển tương ứng. Đây là một mô hình có tính kỷ luật cao và giai đoạn tiếp theo chỉ bắt đầu sau khi hoàn thành giai đoạn trước. * Với V model thì công việc test được tham gia ngay từ đầu. |
| **Ứng dụng**   * Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn. | **Ứng dụng**   * Yêu cầu được xác định rõ ràng. * Xác định sản phẩm ổn định. * Công nghệ không thay đổi và được hiểu rõ bởi nhóm dự án. * Không có yêu cầu không rõ ràng hoặc không xác định. * Dự án ngắn. |
| **Ưu điểm**   * Tốt cho các hệ phần mềm quy mô lớn. * Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa. * Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn. | **Ưu điểm**   * Đây là một mô hình có tính kỷ luật cao và các giai đoạn được hoàn thành cùng một lúc. * Hoạt động tốt cho các dự án nhỏ, khi các yêu cầu được hiểu rất rõ. * Đơn giản và dễ hiểu và dễ sử dụng, dễ quản lý. |
| **Nhược điểm**   * Manager cần có kỹ năng tốt để quản lý dự án, đánh giá rủi ro kịp thời. * Chi phí cao và mất nhiều thời gian để hoàn thành dự án. * Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro. | **Nhược điểm**   * Khó quản lý kiểm soát rủi ro, rủi ro cao. * Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng. * Mô hình kém cho các dự án dài và đang diễn ra. * Không thích hợp cho các dự án có nguy cơ thay đổi yêu cầu trung bình đến cao. |