**Tổng quan về mô hình thác nước (Waterfall model)**

1. **Khái niệm**

Waterfall model (mô hình thác nước) hay còn được gọi là mô hình tuần tự tuyến tính được Winston Royce giới thiệu vào năm 1970. Nó được gọi là mô hình tuần tự tuyến tính là vì các pha của mô hình thông thường chỉ được thực hiện một lần và không có sự quay lại giữa các pha và phải đi theo trình tự từ trên xuống dưới trông như thác nước đổ, tuy nhiên ta vẫn có thể thực hiện quay lại giữa các pha nhưng điều đó là rất ít khi, nếu quay lại giữa các pha sẽ phát sinh thêm chi phí và gây ra nhiều khó khăn. Pha tiếp theo chỉ được bắt đầu khi pha trước đó đã được hoàn thành. Chỉ được quay lại các pha trước đó khi đã hoàn thành xong pha cuối cùng, tức phần mềm đã được hoàn thành.

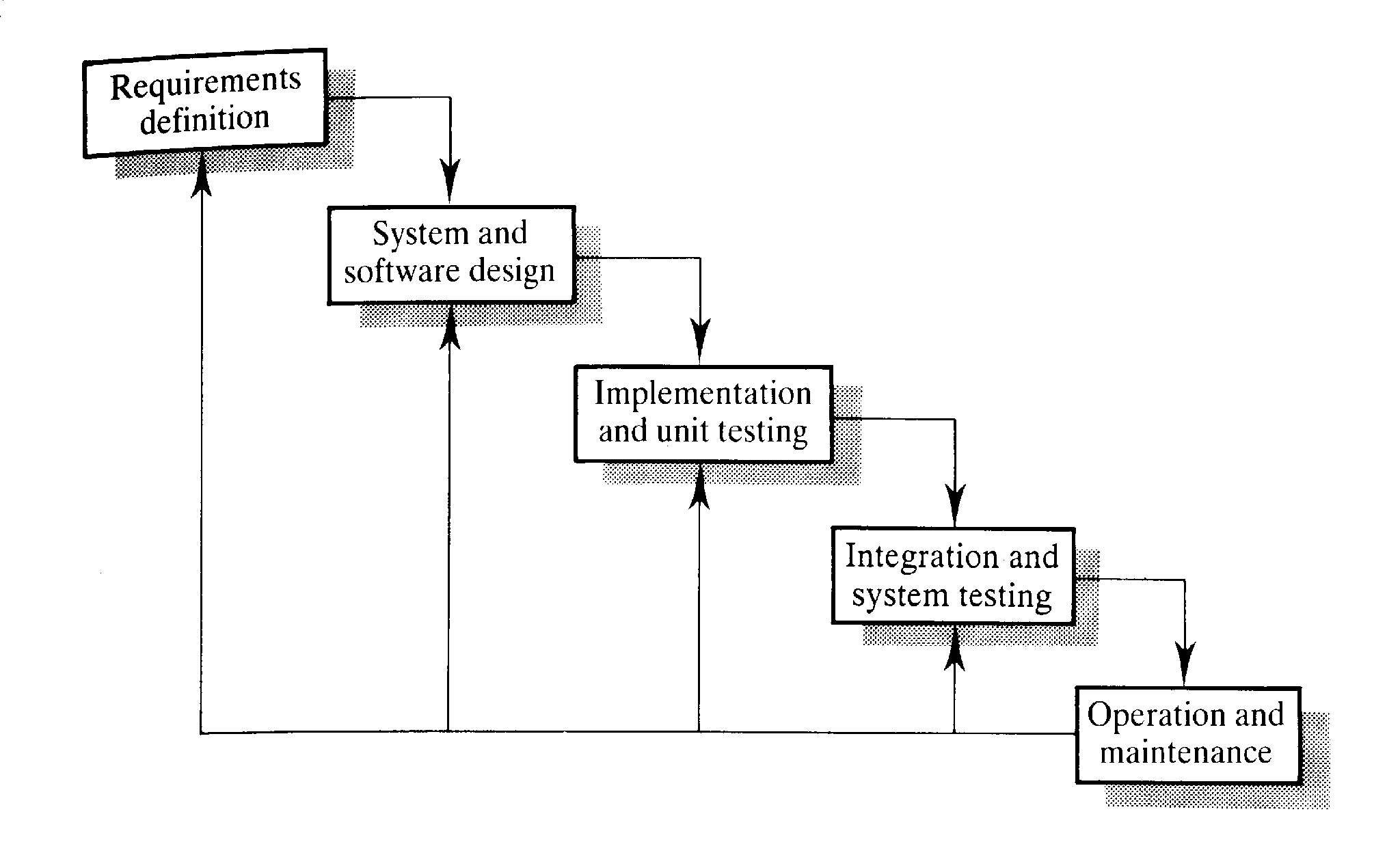
Trước đây mô hình thác nước được sử dụng rất phổ biến nhưng ngày nay nó rất ít khi được sử dụng bởi tính kém linh hoạt giữa các giai đoạn trong quá trình phát triển phần mềm và đòi hỏi phải có sự chặt chẽ, rõ ràng về yêu cầu ngay từ giai đoạn đầu tiên của quá trình phát triển. Mặc dù độ phổ biến của mô hình thác nước không còn được rộng rãi như trước nhưng nó vẫn rất quan trọng vì tất cả các mô hình phát triển phần mềm khác đều dựa trên mô hình thác nước.

1. **Phân tích các giai đoạn**

Mô hình thác nước được chia thành 5 giai đoạn:

1. Phân tích yêu cầu (Requirements definition)
2. Thiết kế hệ thống và phần mềm (System and software design)
3. Hiện thực và kiểm tra moduls (Implementation and unit testing)
4. Tích hợp và kiểm tra hệ thống (Integration and system testing)
5. Chuyển giao và bảo trì (Operation and maintenance)

Hình ảnh minh họa mô hình thác nước:



1. **Phân tích yêu cầu (Requirements definition)**

Người dùng thường chỉ đưa ra những ý tưởng, nhiều khi rất mơ hồ về phần mềm mà họ mong muốn xây dựng. Mục đích của giai đoạn này là để hiểu các yêu cầu chính xác của khách hàng. Vậy nên các lập trình viên sẽ phải thu thập tất cả các yêu cầu cần thiết từ khách hàng, sau đó sẽ phân tích để hiểu đúng những yêu cầu về phần mềm. Yêu cầu của phần mềm là tất cả các yêu cầu về phần mềm do người dùng nêu ra bao gồm các chức năng của phần mềm, hiệu năng của phần mềm, giao diện của phần mềm và một số các yêu cầu khác. Tất cả những chi tiết, yêu cầu của khách hàng sẽ được ghi chép trong tài liệu SRS (tài liệu đặc tả phần mềm)

1. **Thiết kế hệ thống và phần mềm (System and software design)**

**Giai đoạn** này nhằm mục đích chuyển đổi các yêu cầu thu thập được từ khách hàng được ghi chép trong tài liệu SRS thành các bản vẽ thiết kế chi tiết cũng như kiến trúc phần mềm tổng thể của hệ thống.  Tất cả công việc này được ghi lại dưới dạng tài liệu thiết kế phần mềm SDD.

1. **Hiện thực và kiểm tra moduls (Implementation and unit testing)**

Trong giai đoạn này, thiết kế được thực hiện. Nếu SDD được ghi chép hoàn chỉnh, giai đoạn thực hiện sẽ diễn ra suôn sẻ, bởi vì tất cả thông tin mà các nhà phát triển phần mềm cần được chứa trong SDD. Từ đó các lập trình viên sẽ tạo ra chương trình bằng ngôn ngữ lập trình. Các đoạn code sẽ được kiểm tra kỹ lưỡng và sửa đổi nếu có lỗi phát sinh. Các đoạn code nhỏ được thử nghiệm một cách cô lập ban đầu.

1. **Tích hợp và kiểm tra hệ thống (Integration and system testing)**

Ở phần này, các lập trình viên sẽ tích hợp code và kiểm tra sự tương tác của toàn bộ hệ thống để xem các đơn vị có vận hành trơn tru hay không, nếu có trục trặc thì sẽ tiến hành sữa lỗi.

Giai đoạn này rất quan trọng vì chất lượng của phần mềm có tốt hay không là do chất lượng của quá trình kiểm tra. Phần mềm tốt sẽ dẫn đến khách hàng hài lòng.

1. **Chuyển giao và bảo trì (Operation and maintenance)**

Ở phần cuối, sau khi chắc chắn đã sửa hết lỗi và vận hành trơn tru, các lập trình viên sẽ tiến hành chuyển giao cho khách hàng để đưa vào sử dụng, chạy hệ thống trong môi trường thực tế.

Công đoạn bảo trì sữa chữa – nâng cấp sẽ được thực hiện nếu có lỗi phát sinh trong quá trình sử dụng hoặc nếu muốn nâng cao năng suất hệ thống thì các lập trình viên sẽ dựa trên yêu cầu của khách hàng để tiến hành bảo trì nâng cấp, cải tiến phần mềm.