ウォータフォール型は最も代表的で伝統的な開発モデルです。

Mô hình thác nước là mô hình phát triển truyền thống và tiêu biểu nhất

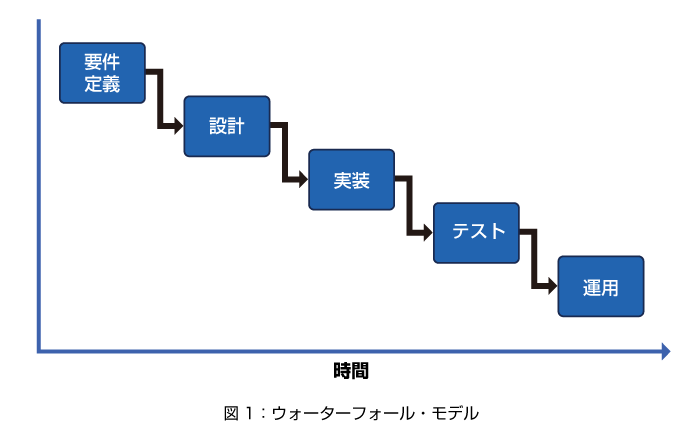
ウォータフォール型では、開発は時系列に沿って段階的に進むこと考え、まず分析や設計など抽象的な工程を行い、それが完了したら次にプログラミングやテストなど具体的な工程を行います。

Mô hình thác nước được được cho rằng tiến hành theo tính bậc thang bám theo dòng chảy thời gian, đầu tiên tiến hành các công đoạn trừu tượng như là thiết kế và phân tích, sau khi hoàn thành thì tiến hành các công đoạn cụ thể như là lập trình và kiểm thử.

各工程は、基本的に後戻りせずに順番に進めていきます。この工程の進む様子が、ちょうど川の上流に降りった雨水が後戻りすることなく、下流に向かって流れ落ちていく姿に似ていることから、ウォータフォール型と名付けられました。

Với mỗi công đoạn, cơ bản thí được tiến hành theo trình tự và không quay lại. Trạng thái, dáng vẻ công đoạn đó, vì giống với hình dạng của dòng sông chảy từ thượng lưu xuống hạ lưu, không quay lại, nên được gọi tên là mô hình thác nước.

定義によって多少の違いはありますが、ウォータフォール型での一般的な工程の進み方は図１．５ようになります。Đối với định nghĩa có khác ít nhiều nhưng mà cách tiến hành công đoạn của mô hình thác nước giống như hình 1.5



要件定義は、顧客が求めている機能や性能を引き出し、システムが備えるべき要件として定義することを意味します。外部設計は、定義された要件に沿って、システムが持つべき機能を定めます。内部設計は、外部設計で定義した機能の具体的な実現方法を定めます。

Yêu cầu của hệ thống có nghĩa là việc mô tả yêu cầu của hệ thống cần chuẩn bị, đưa ra những tính năng, chức năng mà khách hàng yêu cầu. Bản thiết kế sơ bộ là dựa theo những yêu cầu được mô tả, xác định những chức năng mà hệ thống cần phải có. Bản thiết kế chi tiết xác định phương pháp thực hiện chi tiết các chức năng đã mô tả ở bản thiết kế sơ bộ.

製造では、内部設計に沿ってプログラムを製造し、テストでは、製造したプログラムが設計通りにするかどうかを確認し、不具合があれば修正します。そして運用では、完成したシステムを稼働させ、必要に応じて不具合の修正や簡易な機能の追加を行います。

Lập trình, chế tạo là viết chương trình dựa theo bản thiết kế chi tiết, kiểm thử là xác nhận xem chương trình được viết ra có hoạt động theo như thiết kế hay không, nếu có lỗi thì sửa. Hoạt động là vận hành được chương trình đã hoàn thành, tiến hình bổ sung chức năng đơn giản và chỉnh sửa lỗi để phù hợp với yêu cầu.

ウォータフォール型の、段階を追って具体的な型にしていくという考え方は、ものづくり普遍的なアプローチであり、わかりやすいものです。そのため、現在でもウォータフォール型の開発プロセスは広く採用されています。

Có những nhận định về mô hình thác nước như là thực hiện quy trình phát triển dựa theo bậc thang thì sản phẩm tạo ra sự tiếp cận mang tính tổng quát và dễ hiểu. chính vì thế, cho đến hiện nay quy trình phát triển của mô hình thác nước vẫn được sử dụng rộng rãi.

その一方で、ウォータフォール型に対しては、次のような問題点も指摘されています。 Ngược lại, đối với mô hình thác nước, được chỉ ra những vấn đề như sau đây.

* 不具合が、最後の工程であるテストを行うまで見つからない場合が多い。Trường hợp không tìm thấy lỗi cho đến khi tiến hành kiểm thử là công đoạn cuối cùng thì rất nhiều.
* テストで見つかった不具合の原因が分析や設計にある場合、最初までさかのぼって修正しなければならず、それに関連するそれまでの作業が無駄になる。Trường hợp nguyên nhân của lỗi tìm thấy ở kiểm thử là do phân tích và thiết kế gây ra, thì phải sửa lại từ đầu, mà như vậy thì những nghiệp vụ liên quan đến nó trở nên lãng phí.
* どこかの工程が遅れると、その後の全工程の進行に影響を及ぶ。Công đoạn nào đó mà bị chậm thì sẽ gây ảnh hưởng đến việc tiến hành của toàn bộ công đoạn sau đó

これらの問題点は、現実のシステム開発プロジェクトにおいて、完成予定が大幅に遅れたり、必要な費用が当初予算を大きく上回ったりする原因となります。

Trong dự án phát triển hệ thống thực tế, vừa làm chậm tương đối hoàn thành kế hoạch, vừa vượt quá ngân sách ban đầu, nên những vấn đề này trở thành nguyên nhân.

また、顧客のビジネスはダイナミックにいており、顧客がシステムに求める機能がその開発途中に変わる場合も、現実には多くあります。これを「ムービングターゲット」と言いますが、ウォータフォール型ではムービングターゲットに柔軟に対応できないのが実情です。

Ngoài ra, việc kinh doanh của khách hàng là hoạt động linh hoạt, biến động, cũng có trường hợp khách hàng thay đổi yêu cầu chức năng của hệ thống trong quá trình phát triển, trong thực tế có rất nhiều. Điều này gọi là thay đổi mục tiêu, mô hình thác nước không thể đối ứng linh hoạt theo tình hình thực tế.

このようなウォータフォール型の開発モデルの問題点を補い、より柔軟で効率のよい開発手法として提唱されたのが、スパイラル型、アジャイル型などの開発モデルです。

Để bù đắp cho những vấn đề như đó của mô hình phát triển thác nước, người ta đã đề xướng mô kỹ thuật phát triển hiệu suất và linh hoạt hơn như mô hình xoắn ốc, mô hình Agile.