

Nguyên lý Thiết kế và Kiến trúc Phần mềm

Chương 4: Góc nhìn trong thiết kế kiến trúc

Góc nhìn trong thiết kế phần mềm

- Khái niệm góc nhìn trong thiết kế phần mềm là các cách tiếp cận khác nhau để xem xét và giải quyết các vấn đề trong thiết kế phần mềm.
- Mỗi góc nhìn đề cập đến các khía cạnh khác nhau của phần mềm và đưa ra các giải pháp khác nhau để đảm bảo tính đúng đắn, hiệu quả và dễ bảo trì của phần mềm.

Vai trò của góc nhìn trong thiết kế phần mềm

- Vai trò của góc nhìn trong thiết kế phần mềm là giúp các nhà phát triển phần mềm và kiến trúc sư phần mềm có thể nhìn nhận và xử lý các vấn đề phức tạp trong thiết kế phần mềm một cách toàn diện và hệ thống hơn.
- Việc sử dụng các góc nhìn khác nhau cũng giúp đảm bảo tính linh hoạt và mở rộng của phần mềm trong tương lai, cũng như giảm thiểu rủi ro và chi phí trong quá trình phát triển và bảo trì phần mềm.

Vai trò của góc nhìn trong thiết kế phần mềm

- Việc sử dụng các góc nhìn trong thiết kế phần mềm cũng giúp tạo ra các tài liệu thiết kế phần mềm chi tiết và rõ ràng, giúp các thành viên trong dự án có thể hiểu và làm việc với nhau một cách hiệu quả.

Ưu điểm:

Tăng tính toàn diện và hệ thống hóa trong thiết kế phần mềm: Việc sử dụng các góc nhìn khác nhau giúp đảm bảo rằng các khía cạnh quan trọng của phần mềm được xem xét đầy đủ và có thể tích hợp với nhau một cách hiệu quả.

Giảm thiểu rủi ro: Sử dụng nhiều góc nhìn khác nhau trong thiết kế phần mềm giúp tạo ra một mô hình phần mềm tổng thể chính xác hơn, giảm thiểu rủi ro khi triển khai và bảo trì phần mềm.

Tăng tính mở rộng và dễ bảo trì của phần mềm: Sử dụng các góc nhìn khác nhau giúp đảm bảo tính mở rộng và dễ bảo trì của phần mềm trong tương lai.

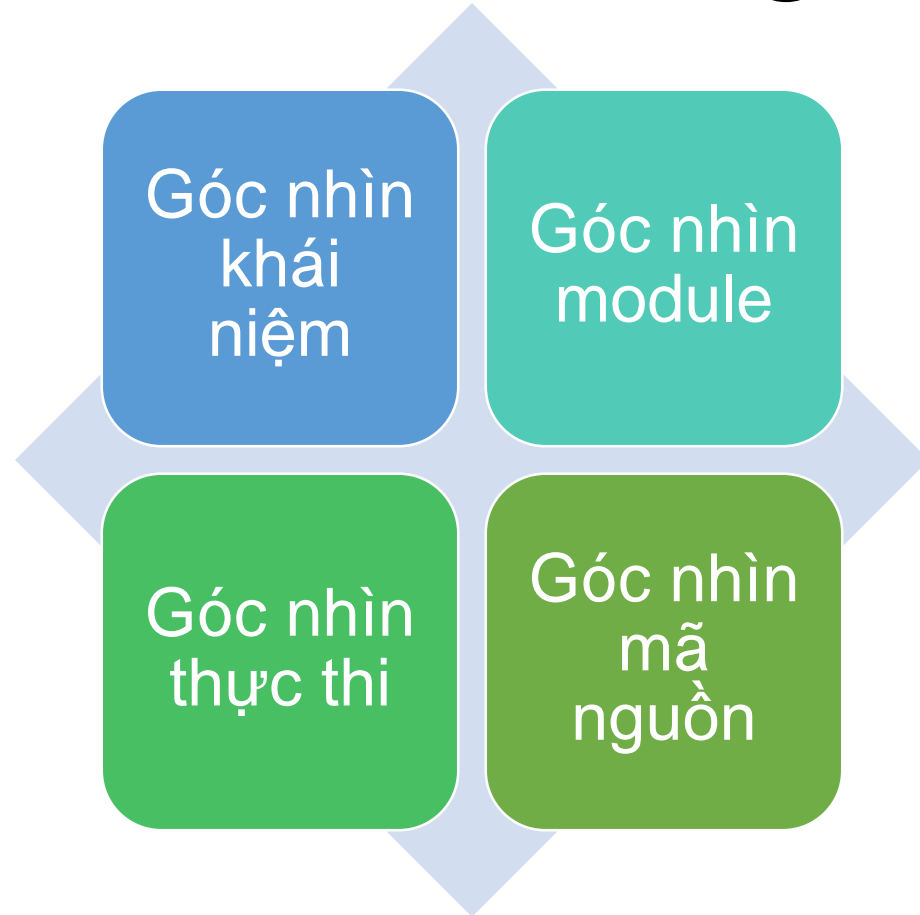
Hạn chế:

Phân chia góc nhìn thiết kế có thể dẫn đến việc thiếu tương tác và hiểu biết giữa các thành viên trong dự án phát triển phần mềm.

Sử dụng quá nhiều góc nhìn khác nhau có thể dẫn đến việc phức tạp hóa quá trình thiết kế phần mềm và làm mất thời gian, tài nguyên và chi phí.

Nếu không sử dụng đúng cách, phân chia góc nhìn thiết kế có thể dẫn đến sự không nhất quán và khó hiểu trong các tài liệu thiết kế phần mềm.

Các góc nhìn cơ bản trong thiết kế phần mềm bao gồm:





Góc nhìn khái niệm

- Từ góc nhìn này, nhà thiết kế phần mềm tập trung vào việc phân tích và định nghĩa các yêu cầu của khách hàng. Họ sẽ sử dụng các công cụ và kỹ thuật để biến các yêu cầu này thành các khái niệm trừu tượng, bao gồm các đối tượng, thuộc tính và mối quan hệ giữa chúng. Các khái niệm này sẽ hình thành cơ sở để thiết kế các module và các đối tượng trong phần mềm.
-



Góc nhìn khái niệm

- Giả sử bạn đang thiết kế một phần mềm quản lý hàng hóa cho một doanh nghiệp thương mại. Từ góc nhìn khái niệm, bạn sẽ phân tích yêu cầu của khách hàng, bao gồm các chức năng cần có trong phần mềm, các đối tượng (như sản phẩm, khách hàng, nhà cung cấp), các thuộc tính của các đối tượng và mối quan hệ giữa chúng (như một khách hàng có thể mua nhiều sản phẩm, một sản phẩm có thể được cung cấp bởi nhiều nhà cung cấp).
-

Góc nhìn khái niệm

- Sơ đồ lớp (Class diagram), sơ đồ thực thể-liên kết (Entity-Relationship diagram), sơ đồ trạng thái (State diagram) được sử dụng trong góc nhìn khái niệm để biểu diễn các đối tượng, thuộc tính và mối quan hệ giữa chúng.
-

Góc nhìn khái niệm

- Góc nhìn khái niệm: Các nhà phát triển phần mềm, kiến trúc sư phần mềm, quản lý dự án và các chuyên gia nghiên cứu yêu cầu khách hàng sẽ tương ứng với góc nhìn khái niệm để định nghĩa các đối tượng, thuộc tính và mối quan hệ giữa chúng trong phần mềm.

Góc nhìn module

- Từ góc nhìn này, nhà thiết kế phần mềm tập trung vào việc phân tích và thiết kế các module hoặc thành phần trong phần mềm. Các module được xác định bằng cách phân chia phần mềm thành các phần nhỏ hơn, hỗ trợ cho việc phát triển, bảo trì và sửa lỗi phần mềm. Nhà thiết kế sử dụng các kỹ thuật thiết kế module để thiết kế các module đơn giản, dễ sử dụng và dễ tái sử dụng.

Góc nhìn module

- Sau khi phân tích yêu cầu và xác định các khái niệm, bạn sẽ thiết kế các module trong phần mềm của mình. Ví dụ, bạn có thể thiết kế một module quản lý thông tin sản phẩm, một module quản lý thông tin khách hàng và một module quản lý thông tin nhà cung cấp. Các module này có thể được phân chia thành các lớp và giao diện người dùng để cải thiện tính bảo mật, dễ bảo trì và sử dụng lại.

Góc nhìn module

- Sơ đồ gói (Package diagram), sơ đồ lớp (Class diagram), sơ đồ hoạt động (Activity diagram), sơ đồ trình tự (Sequence diagram) được sử dụng để biểu diễn các module trong phần mềm và mối quan hệ giữa chúng.

Góc nhìn module

- Góc nhìn module: Các nhà phát triển phần mềm và kiến trúc sư phần mềm sẽ tương ứng với góc nhìn module để xác định các module trong phần mềm và mối quan hệ giữa chúng.

Góc nhìn thực thi

- Từ góc nhìn này, nhà thiết kế phần mềm tập trung vào việc thiết kế các đối tượng, module và chức năng phần mềm để đảm bảo rằng chúng có thể hoạt động đúng cách trong môi trường thực tế. Các kỹ thuật kiểm thử phần mềm và kiểm tra tính toàn vẹn của dữ liệu cũng được sử dụng để đảm bảo rằng phần mềm hoạt động đúng cách và đáp ứng được các yêu cầu của khách hàng.

Góc nhìn thực thi

- Sau khi thiết kế các module, bạn sẽ tiến hành triển khai phần mềm để đảm bảo rằng chúng hoạt động đúng cách trong môi trường thực tế. Ví dụ, bạn có thể kiểm tra tính toàn vẹn của dữ liệu bằng cách nhập một số sản phẩm mới vào phần mềm và đảm bảo rằng chúng được lưu trữ và hiển thị đúng cách. Bạn cũng có thể kiểm tra tính toàn vẹn của phần mềm bằng cách chạy các kịch bản kiểm thử và phát hiện và sửa lỗi.

Góc nhìn thực thi

- Sơ đồ lớp (Class diagram), sơ đồ hoạt động (Activity diagram), sơ đồ trình tự (Sequence diagram), sơ đồ lược đồ (Deployment diagram) được sử dụng để biểu diễn cách phần mềm hoạt động trong môi trường thực tế.

Góc nhìn thực thi

- Góc nhìn thực thi: Các nhà phát triển phần mềm, kiến trúc sư phần mềm và kiểm thử viên sẽ tương ứng với góc nhìn thực thi để đảm bảo phần mềm hoạt động đúng cách trong môi trường thực tế.



Góc nhìn mã nguồn

- Từ góc nhìn này, nhà thiết kế phần mềm tập trung vào việc viết và sắp xếp mã nguồn phần mềm để đảm bảo tính bảo mật, dễ bảo trì và hiệu suất của phần mềm. Các kỹ thuật lập trình được sử dụng để viết mã nguồn phần mềm đảm bảo rằng nó đáp ứng được các yêu cầu của khách hàng và có thể bảo trì và nâng cấp dễ dàng.
-



Góc nhìn mã nguồn

- Sau khi triển khai phần mềm, bạn sẽ viết mã nguồn của phần mềm để đảm bảo tính bảo mật, dễ bảo trì và hiệu suất của phần mềm. Ví dụ, bạn có thể sử dụng các kỹ thuật lập trình như OOP (Object-Oriented Programming) để tạo ra các đối tượng trong phần mềm và giảm thiểu các lỗi có thể xảy ra. Bạn cũng có thể tối ưu mã nguồn của mình để đảm bảo tốc độ thực thi của phần mềm tốt nhất
-



Góc nhìn mã nguồn

- Sơ đồ lớp (Class diagram), sơ đồ hoạt động (Activity diagram), sơ đồ trình tự (Sequence diagram), sơ đồ kế thừa (Inheritance diagram) được sử dụng để biểu diễn các đối tượng, lớp và mối quan hệ giữa chúng trong mã nguồn phần mềm.
-

Góc nhìn mã nguồn

- Góc nhìn mã nguồn: Các nhà phát triển phần mềm sẽ tương ứng với góc nhìn mã nguồn để lập trình phần mềm và triển khai các đối tượng, lớp và mối quan hệ giữa chúng trong mã nguồn phần mềm.