Giai đoạn 4
Thực hiện hệ thống

Chương 10
Phát triển ứng dụng

Nội dung

- Các hoạt động và công việc chính được hoàn tất trong giai đoạn thực hiện hệ thống.
- Vai trò của người phân tích hệ thống trong quá trình phát triển ứng dụng.
- Tầm quan trọng của bảo đảm chất lượng và vai trò của công nghệ phần mềm trong việc phát triển phần mềm.
- Thiết kế từ trên xuống và thiết kế đơn thể.
- Độ kết dính (*cohesion*) và độ kết nối (*coupling*).

Nội dung

- Sử dụng sơ đồ tiến trình và mã giả để lập tài liệu cho luận lý của chương trình.
- Quá trình lập trình và phát sinh mã lệnh.
- Phát triển ứng dụng hướng đối tượng và các ưu điểm của nó.
- Các giai đoạn kiểm tra: kiểm tra đơn thể, kiểm tra sự tích hợp và kiểm tra hệ thống.
- Các loại tài liệu mà người phân tích hệ thống phải chuẩn bị.
- Sự chấp thuận của ban quản lý.

Giới thiệu

- Trong giai đoạn thực hiện hệ thống, nhóm phát triển sử dụng bản mô tả thiết kế hệ thống như là một kế hoạch để xây dựng hệ thống mới.
- Người phân tích và người lập trình có các vai trò khác nhau trong quá trình phát triển ứng dụng.
- Công việc chính của người phân tích là trình bày các phần mô tả một cách rõ ràng, chính xác cho người lập trình.

Các giai đoạn của SDLC

- **Giai đoạn 4: Thực hiện hệ thống (*systems implementation*)**
- **Các công việc**
  - **Phát triển ứng dụng, bao gồm thiết kế, lập trình, kiểm tra và lập tài liệu cho các chương trình và đơn thể.**
  - **Cài đặt và đánh giá, bao gồm đào tạo người sử dụng, chuyển đổi tập tin, thay đổi hệ thống và đánh giá các kết quả.**

Bảo đảm chất lượng

- Bảo đảm chất lượng là vô cùng quan trọng trong các lãnh vực nghiệp vụ, kể cả các chức năng CNTT.
- Mục tiêu chính của bảo đảm chất lượng là phát hiện và tránh các vấn đề càng sớm càng tốt.
- Bảo đảm chất lượng có thể phát hiện:
  - Các yêu cầu không chính xác.
  - Các lỗi sai của thiết kế và lập trình.
  - Tài liệu không chính xác.
  - Việc kiểm tra không hiệu quả.

Bảo đảm chất lượng

- **Công nghệ phần mềm**
  - ***software engineering***
  - **Nhấn mạnh đến chất lượng trong thiết kế phần mềm.**
    - **Thiết kế tốt.**
    - **Cấu trúc hiệu quả.**
    - **Lập tài liệu chính xác.**
    - **Kiểm tra cẩn thận.**
  - ***Software Engineering Institute* (SEI)**
    - **Nhiệm vụ là cải tiến chất lượng của các hệ thống dựa trên phần mềm.**

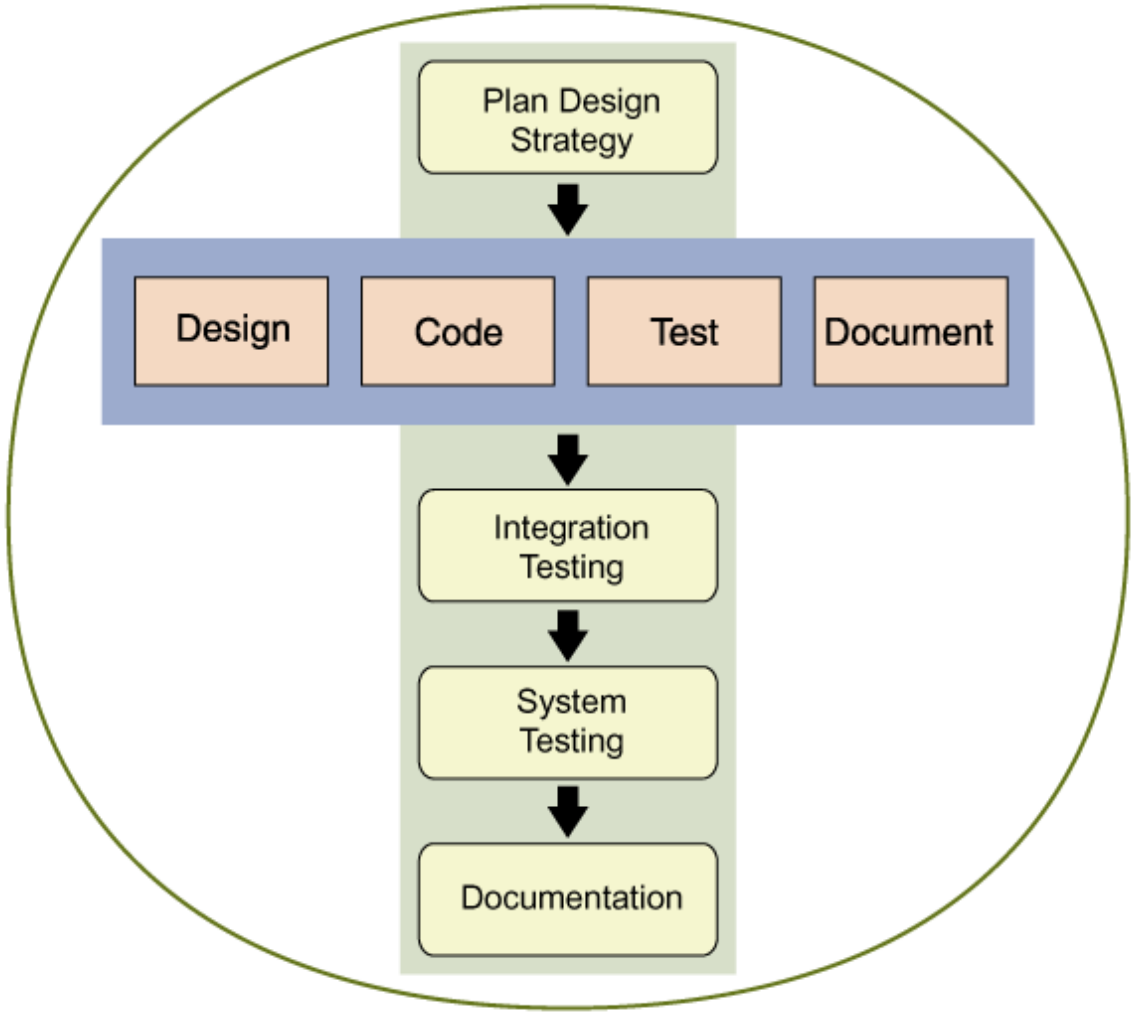
Bảo đảm chất lượng

- ***International Organization for Standardization* (ISO)**
  - **Đưa ra sự thống nhất chung của điều làm cho thực tiễn quản lý được tốt.**
  - **ISO 9000-3 cung cấp tiêu chuẩn bảo đảm chất lượng cho việc phát triển và bảo trì phần mềm.**

Bảo đảm chất lượng

- **Lập chiến lược thiết kế toàn bộ**
  - Sử dụng cách tiếp cận từ trên xuống (đơn thể) và phân chia hệ thống thành nhiều hệ thống con và các đơn thể.
- **Viết các chương trình và đơn thể**
  - Thiết kế, lập trình, kiểm tra và lập tài liệu.
- **Kiểm tra hệ thống**
  - Kiểm tra kết nối các chương trình.
  - Kiểm tra hệ thống.
  - Hoàn thành tất cả các tài liệu.

Bảo đảm chất lượng



Hình 10.1. Các bước chính của phát triển ứng dụng.

Tổng quan về phát triển ứng dụng

- **Xem lại tài liệu**
  - ***documentation review***
  - **Thiết kế chương trình được dựa vào:**
    - Mô tả thiết kế hệ thống (*system design specification*)
    - Các DFD
    - Các mô tả quá trình (*process description*)
    - Các mô hình đối tượng (*object model*)
    - Các sơ đồ lớp (*class diagram*)
    - Các ERD
    - Các dạng trình bày màn hình (*screen layout*)
    - Các dạng trình bày bản báo cáo (*report layout*)
    - Tài liệu nguồn (*source document*)
    - Các mục trong từ điển dữ liệu (*data dictionary entry*)

Phát triển ứng dụng có cấu trúc

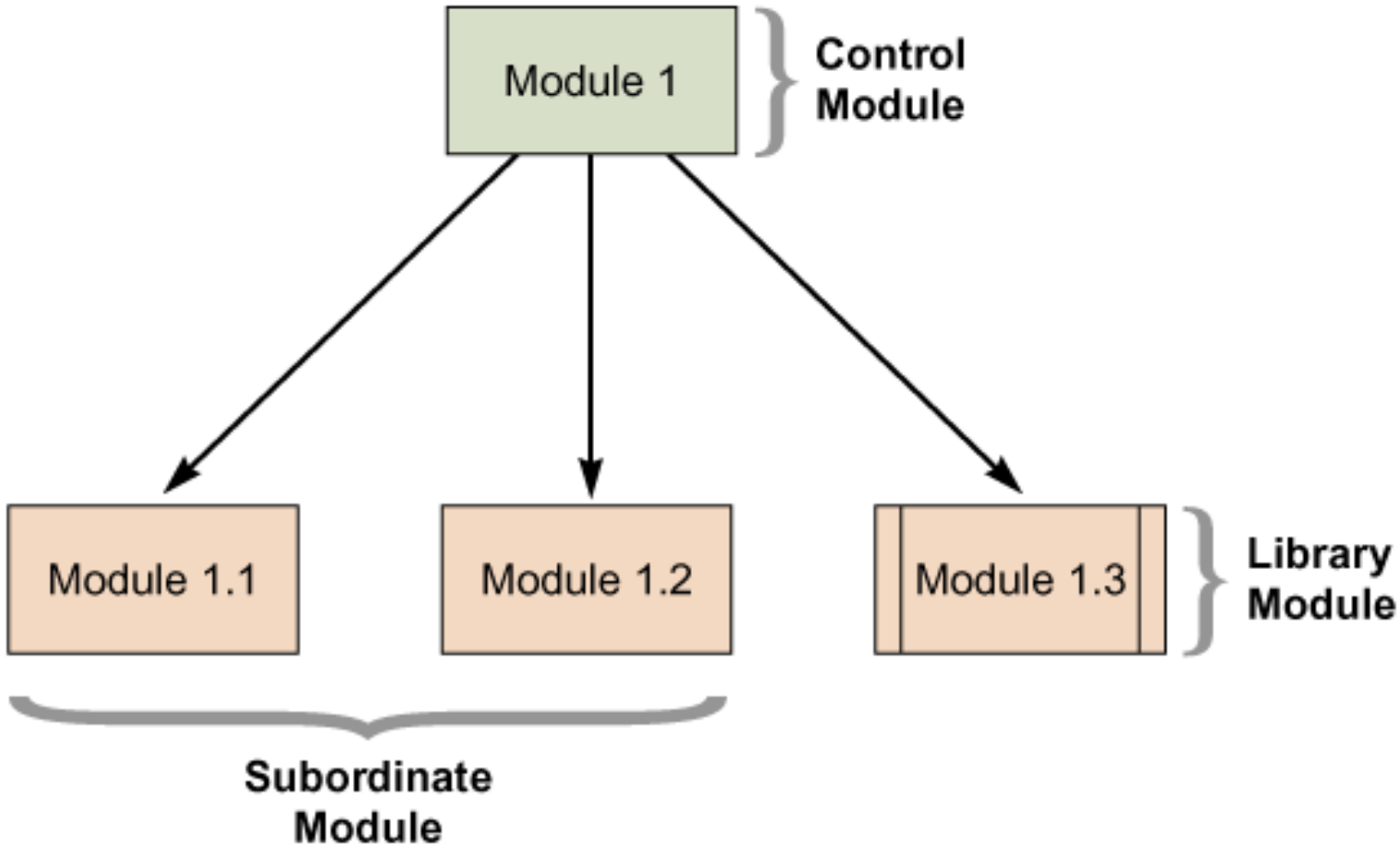
- Người lập trình tạo các đơn thể để thực hiện các chức năng hoặc các công việc cụ thể.
  - Một đơn thể bao gồm các mã lệnh liên quan nhau được tổ chức thành các đơn vị nhỏ để dễ hiểu và dễ bảo trì.
- Hầu hết người phân tích sử dụng cách tiếp cận từ trên xuống khi lập kế hoạch hệ thống.

Phát triển ứng dụng có cấu trúc

Sơ đồ cấu trúc

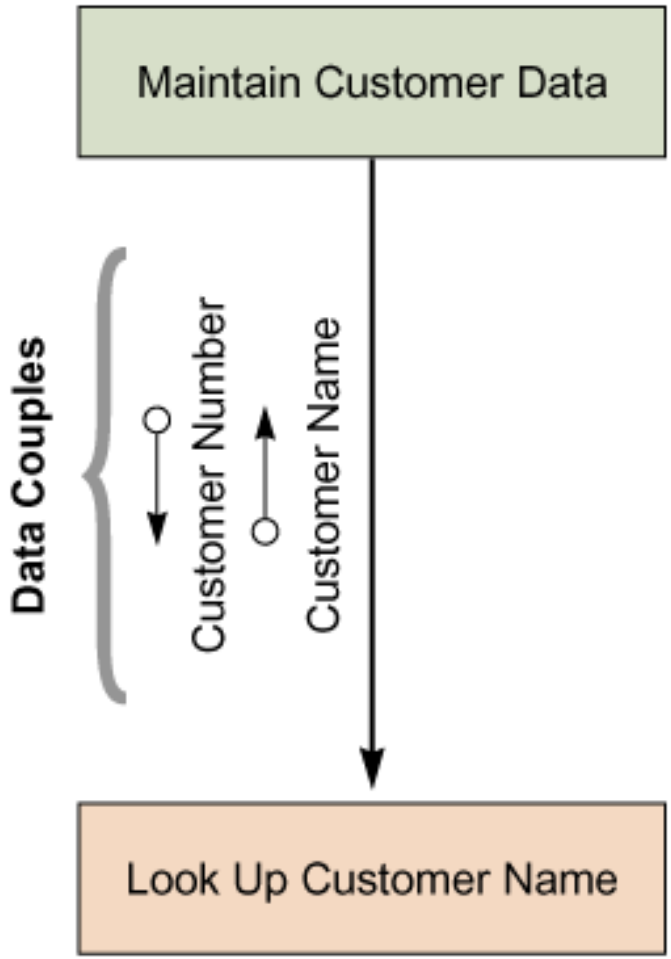
- *structure chart*
- Cho thấy các mối quan hệ giữa các đơn thể chương trình.
- Các ký hiệu của sơ đồ dùng để biểu diễn:
  - Đơn thể (*module*)
  - Kết nối dữ liệu (*data couple*)
  - Kết nối điều khiển (*control couple*)
  - Điều kiện (*condition*)
  - Vòng lặp (*loop*)

Phát triển ứng dụng có cấu trúc



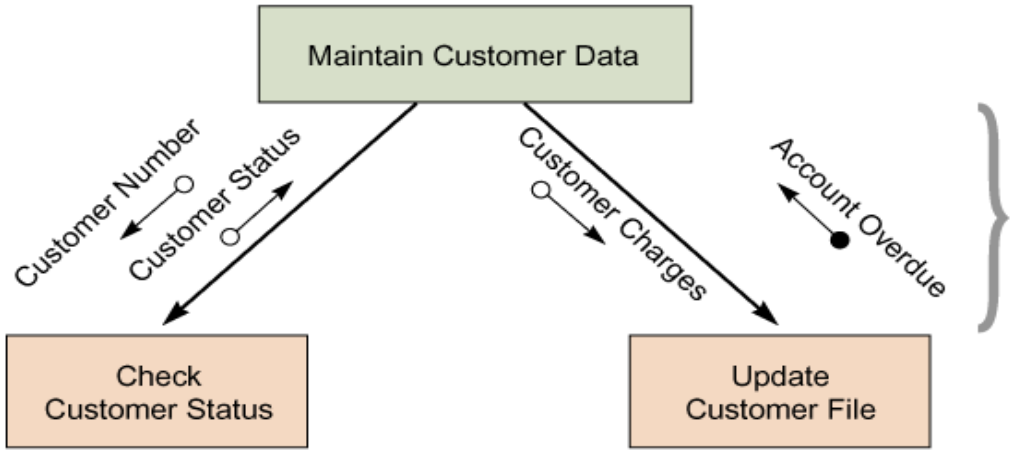
Hình 10.2. Ví dụ về các đơn thể trong sơ đồ cấu trúc.

Phát triển ứng dụng có cấu trúc

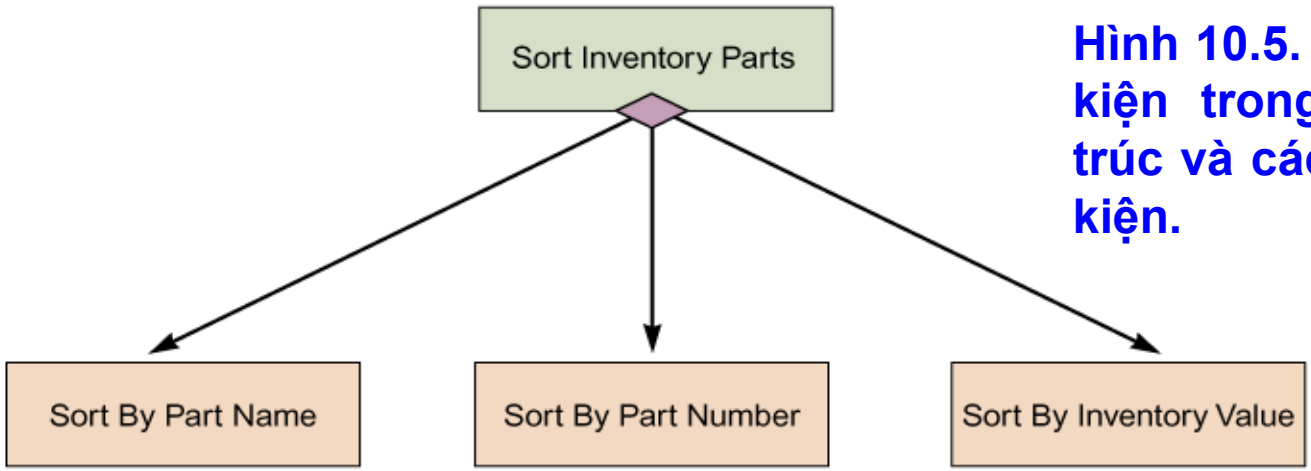


Hình 10.3. Ví dụ về kết nối dữ liệu trong sơ đồ cấu trúc.

Phát triển ứng dụng có cấu trúc

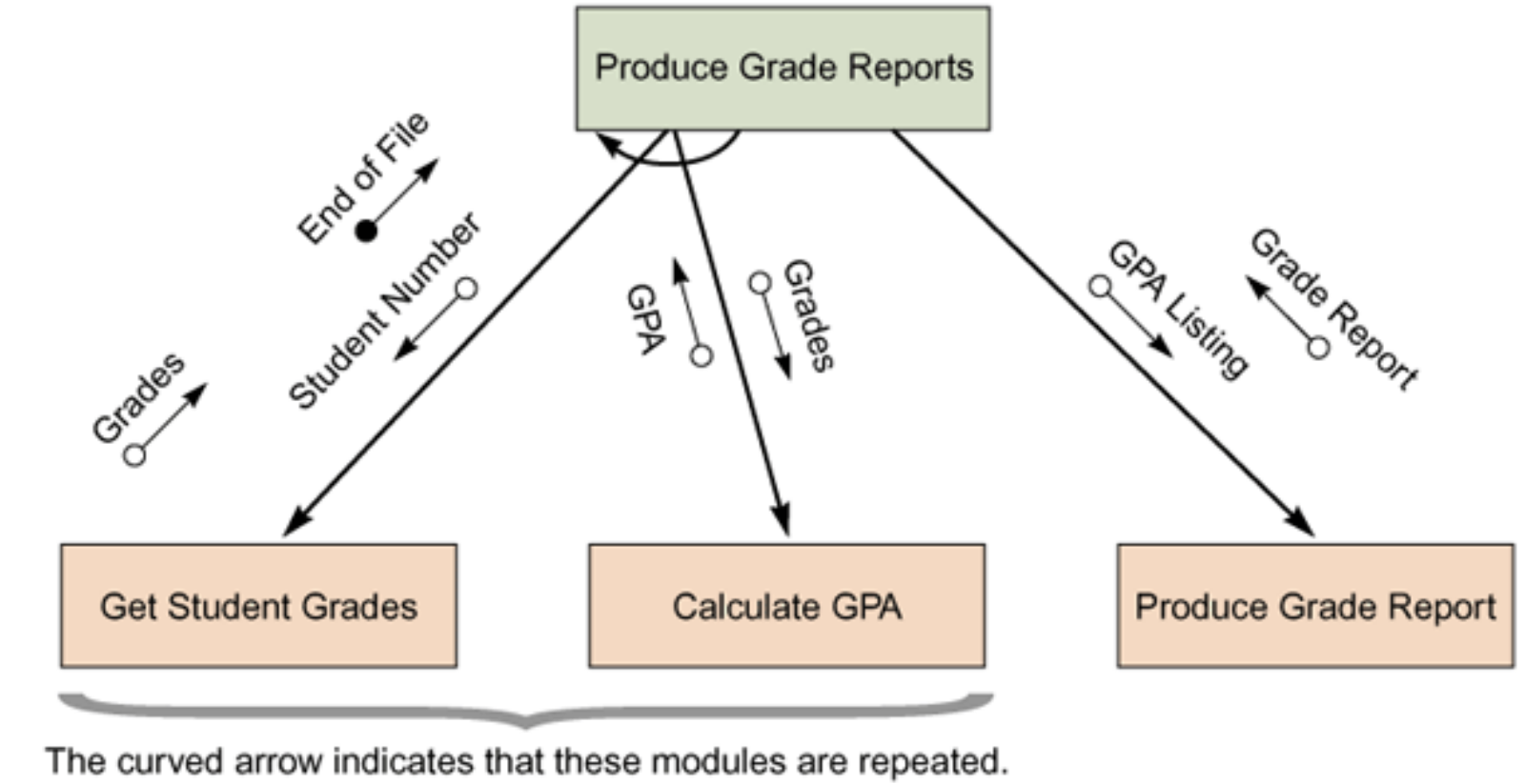


Hình 10.4. Ví dụ về kết nối điều khiển trong sơ đồ cấu trúc.



Hình 10.5. Ví dụ về điều kiện trong sơ đồ cấu trúc và các đường điều kiện.

Phát triển ứng dụng có cấu trúc



Hình 10.6. Ví dụ về vòng lặp trong sơ đồ cấu trúc.

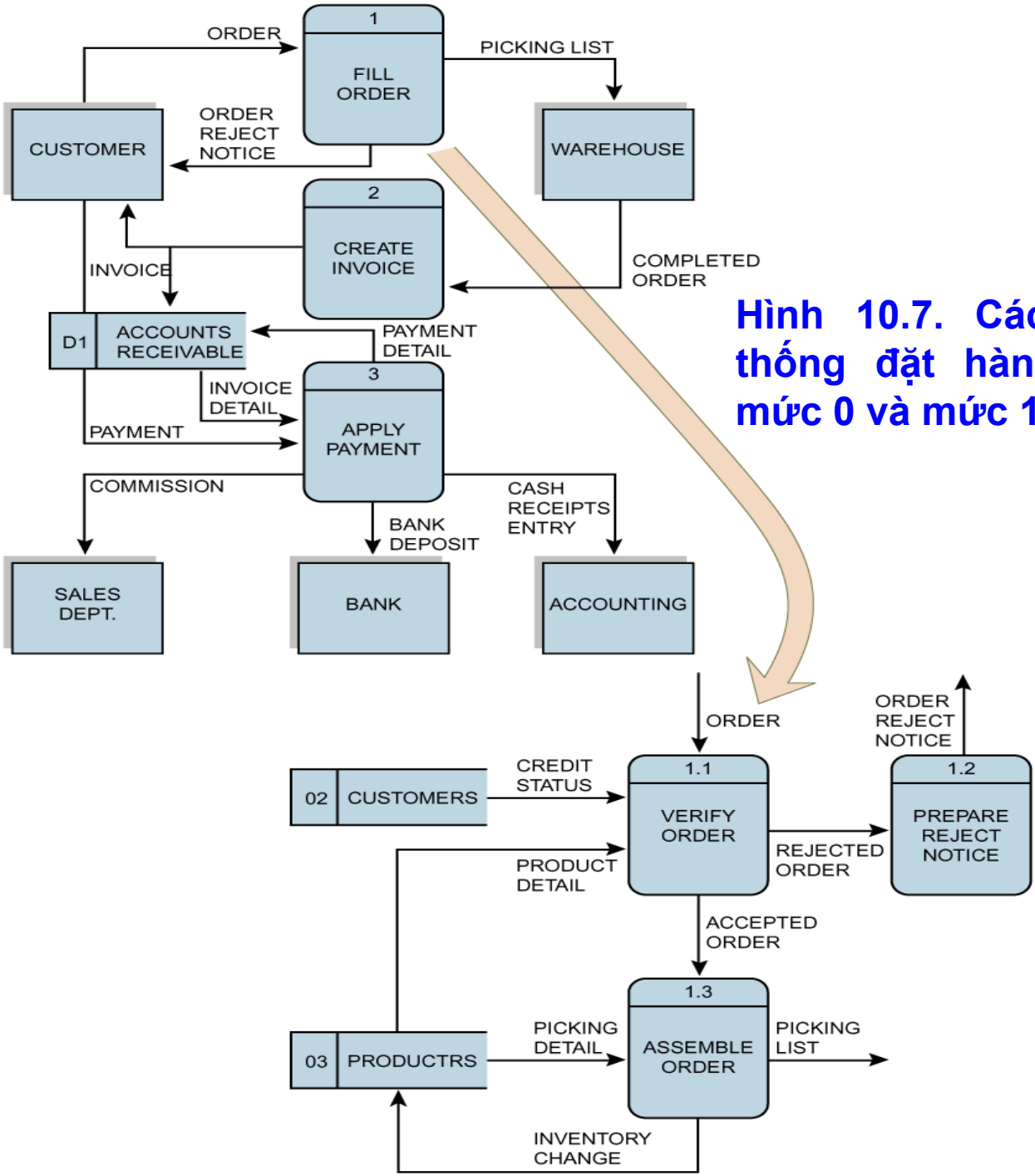
Phát triển ứng dụng có cấu trúc

❖ Độ kết dính và độ kết nối

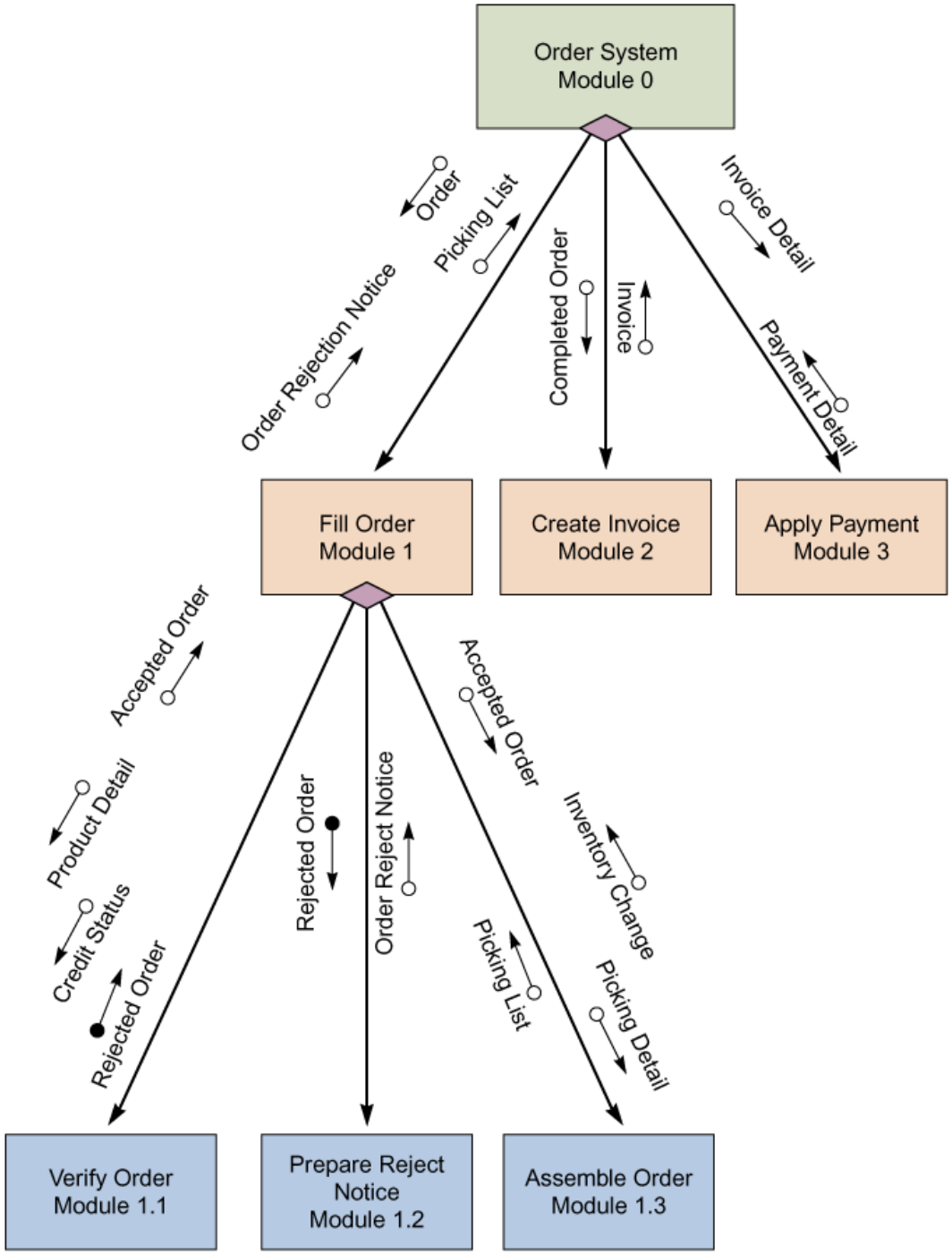
- **Độ kết dính** (*cohesion*) đề cập đến phạm vi của đơn thể và các đặc điểm xử lý.
  - Một đơn thể thực hiện một chức năng đơn lẻ sẽ có độ kết dính cao.
- **Độ kết nối** (*coupling*) đề cập đến các mối quan hệ và sự phụ thuộc lẫn nhau giữa các đơn thể.
  - Các đơn thể độc lập tương đối với nhau sẽ có độ kết nối lỏng lẻo.
  - Nếu một đơn thể tham chiếu đến đoạn chương trình trong một đơn thể khác thì các đơn thể này có độ kết nối chặt chẽ.

Phát triển ứng dụng có cấu trúc

- **Các bước vẽ sơ đồ cấu trúc**
  - Xem lại các DFD và các mô hình đối tượng.
  - Xác định các đơn thể và các mối quan hệ.
  - Thêm vào các kết nối, vòng lặp và điều kiện.
  - Phân tích sơ đồ cấu trúc, các DFD và từ điển dữ liệu.



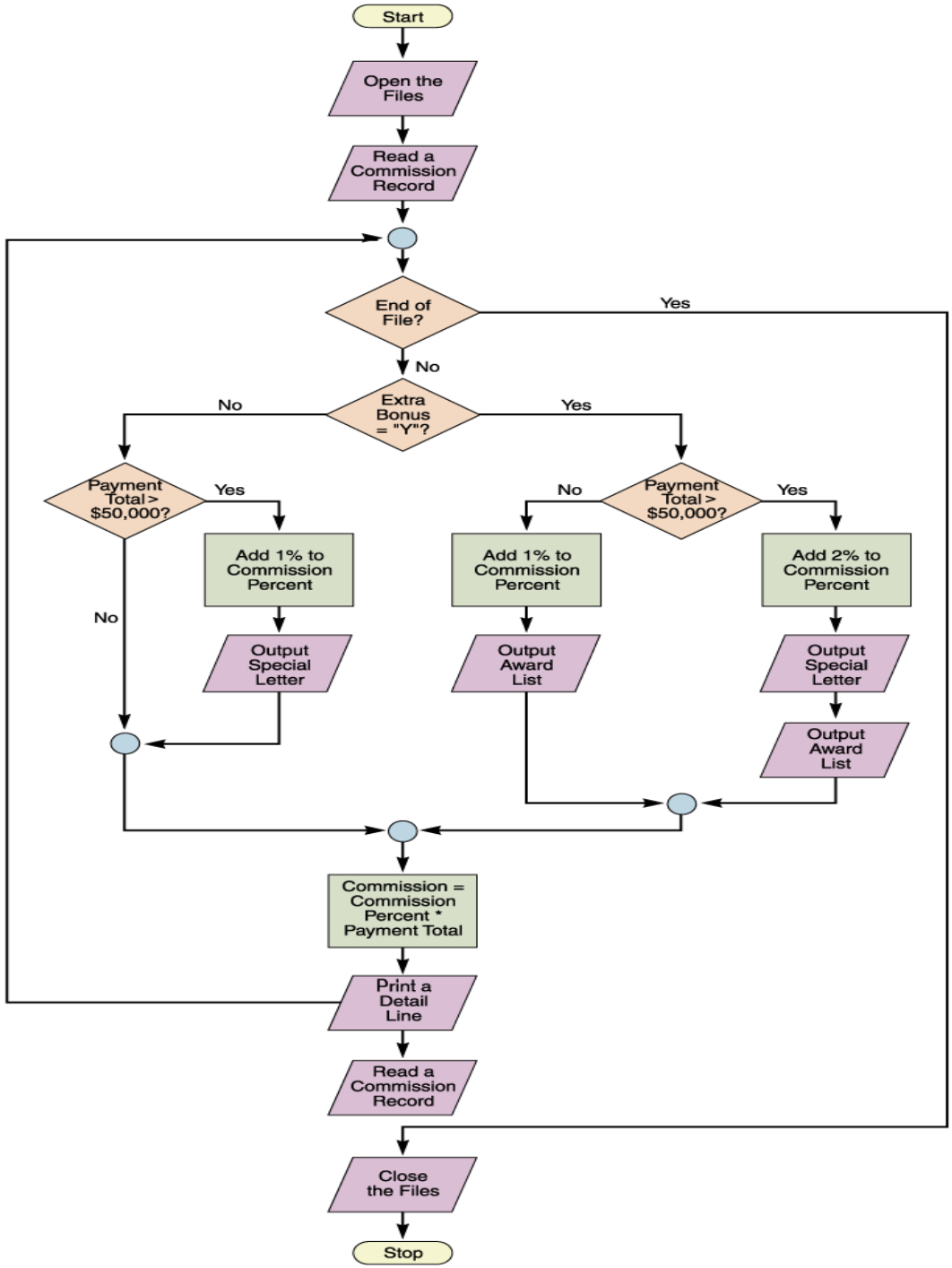
Hình 10.7. Các DFD của Hệ thống đặt hàng. Sơ đồ DFD mức 0 và mức 1.



Hình 10.8. Sơ đồ cấu trúc dựa vào các DFD của Hệ thống đặt hàng.

Các công cụ phát triển ứng dụng

- **Sơ đồ tiến trình của chương trình**
  - ***program flowchart***
  - **Biểu diễn luận lý của đơn thể và sự tương tác giữa các đơn thể bằng đồ họa.**



Hình 10.9. Sơ đồ tiến trình cho thấy cách tính tiền hoa hồng và thực hiện các công việc liên quan.

Các công cụ phát triển ứng dụng

- **Mã giả**
  - *pseudo code*
  - Công cụ để biểu diễn luận lý chương trình.
  - Không phải là một ngôn ngữ cụ thể.

```
Open the files
Read a COMMISSION record
Do until end of file
        If EXTRA BONUS equals Y
                If PAYMENT TOTAL is greater than $50,000
                        Add 2% to COMMISSION PERCENT
                        Output SPECIAL LETTER
                        Output AWARD LIST
                    Else
                        Add 1% to COMMISSION PERCENT
                        Output AWARD LIST
                ENDIF
              Else
                If PAYMENT TOTAL is greater than $50,000
                        Add 1% to COMMISSION PERCENT
                        Output SPECIAL LETTER
                ENDIF
        ENDIF
        Calculate COMMISSION = COMMISSION PERCENT times PAYMENT TOTAL
        Print a detail line
        Read a COMMISSION record
ENDDO
Close the files
End the program
```

Hình 10.10. Ví dụ về mã giả cho thấy các bước luận lý tính tiền hoa hồng trong sơ đồ tiến trình.

Lập trình

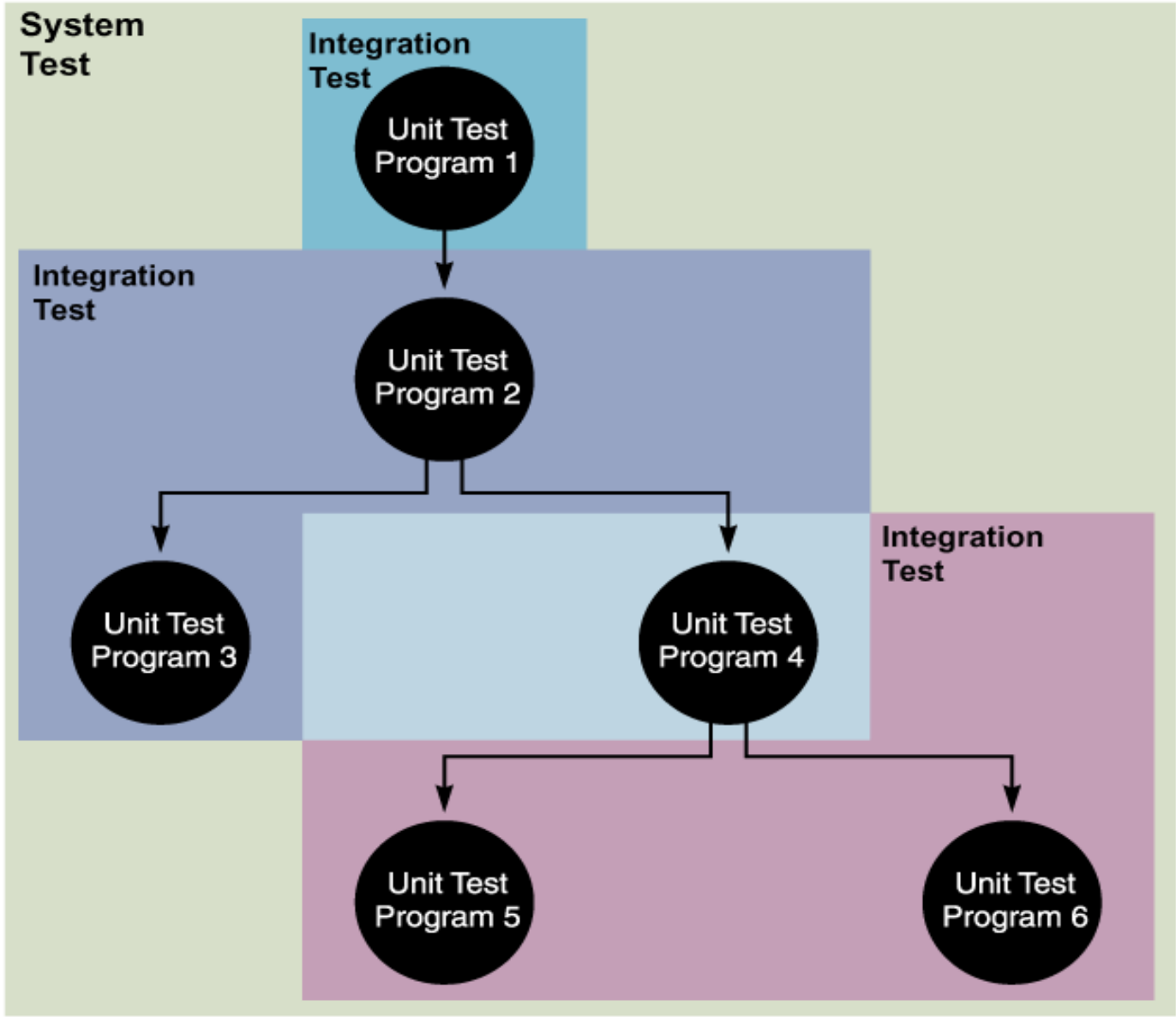
- Quá trình biến đổi luận lý của chương trình thành các lệnh cụ thể có thể được thực hiện bởi hệ thống máy tính.
- Có nhiều ngôn ngữ lập trình.
  - Visual C++
  - Visual Basic
  - SQL
  - HTML
  - Java

Phát triển ứng dụng hướng đối tượng

- Chuyển trực tiếp mô hình đối tượng vào ngôn ngữ lập trình hướng đối tượng.
- Thực hiện thiết kế hướng đối tượng
  - Mối quan hệ và sự tương tác giữa các lớp được mô tả bằng sơ đồ lớp.
  - Mục tiêu chính là chuyển các phương thức của đối tượng thành các đơn thể chương trình và xác định sự kiện hoặc thông điệp nào sẽ kích khởi sự thực hiện của mỗi chương trình.
  - Mỗi sự kiện hoặc thông điệp phải có một hành động tương ứng.

Kiểm tra ứng dụng

- Cần phải kiểm tra để bảo đảm tất cả chương trình đều chạy đúng.
- **Bước 1:** phát hiện lỗi sai ngữ pháp để hoàn tất biên dịch.
- **Bước 2:** loại bỏ các lỗi sai luận lý.
  - Xem lại mã lệnh của chương trình một cách xuyên suốt theo cấu trúc.
- **Bước 3:** kiểm tra
  - Kiểm tra đơn vị chương trình, kiểm tra sự tích hợp và kiểm tra hệ thống.



Hình 10.11. Bước đầu tiên trong việc kiểm tra là kiểm tra đơn vị chương trình, kế tiếp là kiểm tra sự tích hợp, sau đó là kiểm tra hệ thống.

Kiểm tra ứng dụng

- **Kiểm tra đơn vị chương trình**
  - *unit testing*
  - Bao gồm một chương trình riêng biệt.
  - Mục tiêu là xác định và loại bỏ các lỗi sai thực hiện và các lỗi sai luận lý còn lại.
  - Kiểm tra chân là kỹ thuật sử dụng các chân để biểu diễn các điểm vào và điểm ra sẽ được liên kết với chương trình khác hoặc tập tin dữ liệu sau này.

Kiểm tra ứng dụng

Kiểm tra sự tích hợp

- *integration testing*
- Bao gồm hai hoặc nhiều chương trình phụ thuộc lẫn nhau.
- Còn được gọi là kiểm tra liên kết.
- Kiểm tra sự tích hợp bảo đảm các dòng công việc là đúng.
- Cần có dữ liệu kiểm tra để giả lập các điều kiện thực tế và kiểm tra giao tiếp giữa các chương trình.

Kiểm tra ứng dụng

Kiểm tra hệ thống

- *system testing*
- Kiểm tra toàn bộ HTTT và tất cả các tình huống xử lý tiêu biểu.
- Yêu cầu người sử dụng thẩm tra tất cả các tùy chọn xử lý và các kết xuất.
- Sử dụng dữ liệu thật.
- Kiểm tra cuối cùng tất cả các chương trình.
- Bảo đảm có sẵn toàn bộ tài liệu.
- Thẩm tra tất cả các thành phần của hệ thống đều chạy đúng.
- Xác nhận hệ thống có thể giải quyết được khối lượng dữ liệu dự đoán một cách kịp thời và hiệu quả.

Lập tài liệu

- Giải thích hệ thống và hỗ trợ người sử dụng giao tiếp với hệ thống.
- Các loại tài liệu
  - Tài liệu về chương trình.
  - Tài liệu về hệ thống.
  - Tài liệu về vận hành.
  - Tài liệu cho người sử dụng.

Lập tài liệu

- **Tài liệu về chương trình**
  - *program documentation*
  - Bắt đầu trong giai đoạn phân tích hệ thống và tiếp tục trong giai đoạn thực hiện hệ thống.
  - Bao gồm mô tả các quá trình và cách trình bày bản báo cáo.
  - Người lập trình cung cấp tài liệu cùng các ghi chú để dễ hiểu hơn và bảo trì chương trình.
  - Người phân tích phải thẩm tra tài liệu về chương trình là đầy đủ và chính xác.

Lập tài liệu

Tài liệu về hệ thống

- *system documentation*
- Tài liệu về hệ thống mô tả các chức năng của hệ thống và chúng được thực hiện như thế nào.
- Hầu hết tài liệu về hệ thống được chuẩn bị trong các giai đoạn phân tích và thiết kế hệ thống.
- Tài liệu về hệ thống bao gồm:
  - Các mục trong từ điển dữ liệu
  - Các DFD
  - Các mô hình đối tượng
  - Các dạng trình bày màn hình
  - Các tài liệu nguồn
  - Yêu cầu hệ thống ban đầu

Lập tài liệu

Tài liệu về vận hành

- *operations documentation*
- Được sử dụng trong môi trường máy tính lớn hoặc máy tính *mini* với việc xử lý tập trung và lập thời biểu công việc theo bó.
- Tài liệu cho biết các nhóm điều hành CNTT chạy các chương trình khi nào và như thế nào.
- Ví dụ: bản chạy chương trình chứa thông tin cần thiết để xử lý và phân phát kết xuất.

Lập tài liệu

- **Tài liệu cho người sử dụng**
  - *user documentation*
  - **Bao gồm các mục tiêu biểu sau đây:**
    - Tổng quan về hệ thống.
    - Mô tả tài liệu nguồn với các ví dụ tiêu biểu.
    - Các màn hình trình đơn, màn hình nhập dữ liệu.
    - Các bản báo cáo có sẵn với các ví dụ tiêu biểu.
    - Bảo mật và thông tin vết kiểm tra.
    - Trách nhiệm nhập, xuất và xử lý.
    - Các thủ tục giải quyết các thay đổi/vấn đề.
    - Các ví dụ ngoại lệ và các tình huống sai.
    - Các câu thường hỏi (FAQ).
    - Phần trợ giúp giải thích và cập nhật sổ tay.

Lập tài liệu

- Tài liệu cho người sử dụng
  - Tài liệu trực tuyến có thể hướng dẫn người sử dụng và giảm yêu cầu hỗ trợ trực tiếp.
    - Trợ giúp cảm ngữ cảnh.
    - Hướng dẫn theo cách tương tác.
    - Gợi ý và chỉ dẫn.
    - Siêu văn bản.
  - Tài liệu đã viết thường được cung cấp trong sổ tay người sử dụng.
  - Người phân tích chuẩn bị tài liệu và người sử dụng xem lại nó và tham gia vào việc viết sổ tay.

Sự chấp thuận của ban quản lý

- Sau khi hoàn thành kiểm tra hệ thống, các kết quả được trình bày cho ban quản lý.
  - Các kết quả kiểm tra.
  - Tình trạng của tất cả các tài liệu cần có.
  - Dữ liệu của người sử dụng có tham gia.
  - Thời biểu chi tiết, đánh giá chi phí và các yêu cầu của người sử dụng.
- Nếu được chấp thuận, thiết lập thời biểu cài đặt và đánh giá hệ thống.