

## MỤC LỤC

<b>CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM .....</b>	<b>5</b>
1.1 Giới thiệu tổng quan về CNPM .....	5
1.1.1 Công nghệ phần mềm nhìn từ góc độ lịch sử .....	5
1.1.2 Từ góc độ kinh tế .....	8
1.1.3 Khủng hoảng phần mềm .....	8
1.2 Một số khái niệm cơ bản .....	11
1.2.1 Phần cứng (hardware).....	11
1.2.2 Phần mềm (software).....	11
1.2.3 Công nghệ (engineering).....	17
1.2.4 Công nghệ phần mềm (software engineering).....	17
1.2.5 Sự khác biệt giữa công nghệ phần mềm và khoa học máy tính? .....	19
1.2.6 Sự khác nhau giữa kỹ nghệ phần mềm và kỹ nghệ hệ thống .....	19
1.2.7 Tiến trình phần mềm là gì?.....	21
1.2.8 Mô hình quy trình phần mềm là gì?.....	22
1.2.9 Chi phí của công nghệ phần mềm bao gồm những gì? .....	22
1.2.10 Các phương pháp công nghệ phần mềm là gì? .....	24
1.2.11 CASE (Computer-Aided Software Engineering) là gì? .....	25
1.2.12 Thế nào là một phần mềm tốt? .....	25
1.2.13 Những thách thức chính đối với công nghệ phần mềm? .....	26
1.3 Các trách nhiệm đạo đức và nghề nghiệp .....	26
1.3.1 Các vấn đề về trách nhiệm nghề nghiệp.....	26
1.3.2 Tập các chuẩn mực đạo đức .....	26
1.4 Nhân tố con người và sự phân hóa nghề nghiệp trong CNPM .....	27
1.4.1 Nhân tố con người trong ngành công nghệ phần mềm .....	27
1.4.2 Phân loại nghề nghiệp .....	27
<b>CHƯƠNG 2. QUY TRÌNH XÂY DỰNG PHẦN MỀM.....</b>	<b>32</b>
2.1 Mô hình phát triển phần mềm .....	33
2.1.1 Mô hình thác nước (Mô hình tuyến tính - The linear sequential model).....	33
2.1.2 Mô hình mẫu thử (Prototyping model) – Mô hình xây dựng tiến triển .....	34
2.1.3 Mô hình phát triển dựa trên thành phần .....	36
2.1.4 Mô hình xoắn ốc .....	38

2.1.5 Mô hình phát triển ứng dụng nhanh (RAD – Rapid Application Development) .....	39
2.1.6 Mô hình tăng trưởng (incremental model) .....	40
2.1.7 Phát triển các hệ thống hình thức hóa (Formal System Development) .....	40
2.1.8 Phát triển hướng sử dụng lại.....	42
2.2 Các hoạt động trong quy trình phần mềm.....	46
2.2.1 Đặc tả phần mềm.....	47
2.2.2 Thiết kế phần mềm và cài đặt .....	48
2.2.3 Đánh giá phần mềm.....	49
2.2.4 Cải tiến phần mềm.....	50
2.3 Các vấn đề liên quan đến tiến trình phần mềm .....	50
<b>CHƯƠNG 3. ĐẶC TẢ PHẦN MỀM .....</b>	<b>52</b>
3.1 Yêu cầu phần mềm là gì? .....	53
3.2 Yêu cầu hệ thống.....	54
3.2.1 Phân loại yêu cầu hệ thống phần mềm.....	54
3.2.2 Yêu cầu chức năng.....	54
3.2.3 Yêu cầu phi chức năng.....	55
3.2.4 Yêu cầu miền ứng dụng.....	57
3.3 Yêu cầu của người sử dụng.....	57
3.4 Quy trình xác định yêu cầu .....	58
3.4.1 Nghiên cứu tính khả thi dự án .....	58
3.4.2 Phát hiện và phân tích yêu cầu.....	60
3.4.3 Tài liệu đặc tả yêu cầu.....	66
3.4.4 Đánh giá yêu cầu.....	74
3.4.5 Lập kế hoạch quản lý yêu cầu.....	74
<b>CHƯƠNG 4. CÁC MÔ HÌNH HỆ THỐNG.....</b>	<b>76</b>
4.1 Mô hình ngữ cảnh .....	76
4.2 Mô hình ứng xử.....	77
4.2.1 Mô hình luồng dữ liệu.....	77
4.2.2 Mô hình máy trạng thái .....	78
4.3 Mô hình dữ liệu .....	80
4.4 Mô hình đối tượng.....	81
4.4.1 Mô hình thừa kế.....	82

4.4.2 Mô hình kết hợp .....	83
4.4.3 Mô hình ứng xử.....	84
4.5 Phương pháp hướng cấu trúc .....	85
<b>CHƯƠNG 5. THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG .....</b>	<b>86</b>
5.1 Thiết kế hệ thống.....	86
5.1.1 Các hoạt động trong quá trình thiết kế hệ thống.....	86
5.1.2 Thiết kế kiến trúc .....	88
5.1.3 Thiết kế giao diện người dùng .....	95
5.1.4 Thiết kế cấu trúc dữ liệu .....	100
5.1.5 Thiết kế thuật toán/thủ tục .....	101
5.2 Cài đặt hệ thống .....	101
5.2.1 Các yêu cầu đối với lập trình viên.....	101
5.2.2 Quá trình phát triển của kỹ thuật lập trình .....	102
5.2.3 Chọn ngôn ngữ lập trình cho ứng dụng.....	104
5.2.4 Một số nguyên tắc lập trình .....	105
<b>CHƯƠNG 6. KIỂM THỬ PHẦN MỀM.....</b>	<b>106</b>
6.1 Xác minh và thẩm định phần mềm .....	106
6.2 Kiểm thử phần mềm.....	106
6.3 Những nguyên tắc trong kiểm thử phần mềm.....	110
6.4 Quy trình kiểm thử .....	111
6.5 Các mức kiểm thử phần mềm .....	112
6.5.1 Kiểm thử đơn vị (Unit testing) .....	112
6.5.2 Kiểm thử tích hợp (Integration testing) .....	113
6.5.3 Kiểm thử chức năng (Functional testing).....	115
6.5.4 Kiểm thử hệ thống (System testing) .....	115
6.5.5 Kiểm thử tích hợp hệ thống (System integration testing) .....	115
6.5.6 Kiểm thử sự thực thi (Performance testing) .....	115
6.6 Kỹ thuật kiểm thử phần mềm.....	116
6.6.1 Kiểm thử hộp đen.....	116
6.6.2 Kiểm thử hộp trắng.....	117
6.6.3 Kỹ thuật kiểm tra tĩnh và động (static và dynamic).....	118
6.6.4 Kỹ thuật kiểm thử hộp xám .....	118
6.7 Thiết kế Testcase.....	119

6.7.1 Thiết kế testcase cho kiểm thử hộp đen .....	119
6.7.2 Tạo testcase cho phương pháp kiểm thử hộp trắng.....	125
<b>CHƯƠNG 7. BẢO TRÌ PHẦN MỀM.....</b>	<b>136</b>
7.1 Bảo trì phần mềm.....	136
7.1.1 Dự đoán bảo trì.....	137
7.1.2 Dự đoán thay đổi .....	137
7.2 Các quy trình cải tiến phần mềm .....	138
7.3 Tái kỹ nghệ hệ thống (System re-engineering) .....	140
<b>CHƯƠNG 8. TỔNG QUAN VỀ QUẢN LÝ DỰ ÁN PHẦN MỀM .....</b>	<b>142</b>
8.1 Mở đầu .....	142
8.2 Dự án là gì? .....	142
8.2.1 Dự án Công nghệ Thông tin là gì? .....	143
8.2.2 Các đặc trưng của dự án .....	143
8.2.3 Bộ ba ràng buộc: .....	144
8.3 Quản lý Dự án là gì? .....	144
8.4 Các giai đoạn của tiến trình quản trị dự án .....	146
8.5 Vai trò, trách nhiệm của người quản lý dự án.....	148
8.5.1 Vị trí của nhà quản lý dự án trong bối cảnh chung của dự án.....	148
8.5.2 vai trò của nhà quản lý dự án.....	148
8.5.3 Các kỹ năng cần thiết của người quản lý dự án .....	149
8.5.4 Phẩm chất của nhà quản lý dự án .....	150
8.6 Ước lượng dự án .....	150
8.6.1 Ước lượng thời gian các hoạt động.....	150
8.6.2 Ước lượng chi phí dự án.....	150

## CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM

### MỤC ĐÍCH

- *Giới thiệu tổng quan về CNPM*
- *Nắm được một số khái niệm cơ bản*
- *Hiểu được các trách nhiệm đạo đức và nghề nghiệp của người kỹ sư CNPM.*
- *Hiểu được nhân tố con người và sự phân loại nghề nghiệp trong CNPM.*

### 1.1 Giới thiệu tổng quan về CNPM

#### 1.1.1 Công nghệ phần mềm nhìn từ góc độ lịch sử

- Năm 1946 máy tính điện tử ra đời.
- Những năm 1950 máy tính được thương mại hóa: phần mềm bắt đầu được phát triển.

Máy tính điện tử đầu tiên cho mục đích thương mại là máy UNIVAC-1 được sản xuất năm 1951 ở Mỹ. Từ khoảng những năm 1955 thì bắt đầu có các phần mềm, tức là các chương trình máy tính.

- Những năm 1960: gặp những thất bại về phát triển phần mềm.

Cho đến nay, nếu lấy phương pháp và mức độ phức tạp làm căn cứ, thì có thể chia quá trình phát triển phần mềm thành 3 giai đoạn: 1955 - 1970, 1971 - 1985, 1986 đến nay. Tất nhiên sự phân chia này chỉ là tương đối mà thôi.

Đặc điểm của từng giai đoạn là:

- **1955 - 1970:** Tính toán và quản lý rời rạc, quản lý nhỏ. Đặc tả yêu cầu của khách hàng lúc đó còn dùng ngôn ngữ tự nhiên thông thường.

**Ví dụ:** các câu kiểu như: tôi muốn có tệp dữ liệu chứa những thông tin về bán sản phẩm như số hóa đơn, người bán, mặt hàng,...; Dữ liệu sẽ được nhập từ bàn phím, có kiểm tra rồi mới đưa vào tệp; Hàng tháng tôi muốn lấy thông tin về doanh thu, lãi, số hàng bán....

Thiết kế thời ấy không được soạn thảo ra giấy mà chỉ hình thành trong tư duy của người lập trình. Người lập trình vừa viết chương trình vừa suy nghĩ về cách tổ chức dữ liệu, cách sử dụng các thuật toán sao cho chương trình chạy tốt, đáp ứng được yêu cầu của khách hàng.

Giai đoạn này các chương trình nhỏ, tính toán chuyên dụng ( chỉ gồm khoảng trên dưới vài trăm dòng lệnh). Phương pháp lập trình được sử dụng thời đó là **lập trình tuyến tính**, tức là cách viết chương trình trong đó phần lớn các lệnh được đặt theo trình tự thực hiện của chúng, nghĩa là lệnh cần thực hiện trước sẽ được viết trước, lệnh thực hiện sau được viết sau.

Các hoạt động xử lý của chương trình là xử lý số, theo lô,... Ngôn ngữ được sử dụng trong giai đoạn này là ngôn ngữ mã máy, hợp ngữ và chúng mang tính đặc thù theo từng máy.

Các tiêu chí đánh giá gồm tính nhanh, giải được bài toán lớn (dùng bộ nhớ hiệu quả). Các công nghệ giai đoạn này như Bóng điện tử (các tính toán là chậm, bộ nhớ nhỏ), ...

- **1971 - 1985:** Lúc này đã có nhu cầu xây dựng các phần mềm thời gian thực (nghĩa là tính toán và sử dụng ngay kết quả, ví dụ tính toán trong lò phản ứng hạt nhân phải có ngay kết quả để điều khiển kịp thời).

Lúc này có nhu cầu xây dựng các mạng cục bộ và kết nối các cơ sở dữ liệu. Đặc tả thời đó chú trọng vào đặc tả đầu vào, đầu ra; dữ liệu và các luồng dữ liệu (data flow).

**Ví dụ:** những tệp dữ liệu lưu trữ những thông tin gì, trong quá trình xử lý thì dữ liệu được tính toán và di chuyển ra sao. Đầu ra của tệp dữ liệu này sau khi xử lý lại có thể trở thành đầu vào của tệp khác...

Thiết kế thời đó chú trọng tới cấu hình hệ thống (system configuration). Vấn đề cấu trúc đối với dữ liệu và giải thuật cũng được chú ý, vì vấn đề lớn cần phải được **mô tả có cấu trúc** cho dễ hiểu.

Các chương trình tiêu biểu thời đó có thể kể tới **hệ điều hành DOS, UNIX**...Lúc đó cũng đã có những cơ sở dữ liệu có thể truy cập từ xa. Do nhu cầu thực tế, các ngôn ngữ lập trình ra đời giai đoạn này đã có khả năng hỗ trợ phương pháp lập trình có cấu trúc, nghĩa là chương trình được **phân chia thành các chương trình con**, mỗi chương trình con có tên và các chức năng xử lý riêng, có thể giao tiếp với các chương trình con khác thông qua các đối số và kiểu trả về (nếu có).

Ở giai đoạn này phần mềm là thường là sản phẩm đa nhiệm, đa người dùng. Các hoạt động xử lý là xử lý số, ký tự, theo lô và thời gian thực. Giai đoạn này xuất hiện hình thức lưu trữ thông tin trực tuyến (CSDL).

Ngôn ngữ lập trình sử dụng trong giai đoạn này là các ngôn ngữ có cấu trúc như PL1, Algol 60, Fortran, COBOL, ..

Các tiêu chí đánh giá phần mềm gồm tính nhanh, giải được bài toán lớn, nhiều người dùng. Công nghệ: bán dẫn (tính toán nhanh hơn, bộ nhớ khá,...), cơ sở dữ liệu. Yêu cầu bảo trì: Sửa lỗi, thích nghi.

- Từ **1986 đến nay:** Đây là thời kỳ của máy vi tính PC, thời nổi mạng tầm rộng, mạng toàn cầu Internet.

Đặc tả dự án được biết nhiều là hướng đối tượng, công cụ CASE (Computer Aided Software Engineering).

Trong thiết kế người ta chú ý đến môđun (module), đối tượng (object), giao thức (protocol) và giao diện (interface). Giao thức hay giao diện nói về sự trao đổi giữa các đối tượng.

Khi cài đặt trên máy tính người ta thường dùng ngôn ngữ hướng đối tượng, thế hệ 4, lập trình trực quan. Phần mềm lớn, tinh vi, tin cậy, hướng người dùng. Gồm các hệ chuyên gia, trí tuệ nhân tạo, phần mềm nhúng, các dịch vụ Web được sử dụng rộng rãi. Internet ngày càng được mở rộng. Cơ sở dữ liệu hướng đối tượng, kho dữ liệu phát triển.

Ngoài các phần mềm được viết theo đặt hàng, người ta chú trọng đến các phần mềm đóng gói như: Netscape, Internet Explorer, Word, Excel...

Các tiêu chí đánh giá phần mềm gồm: Tính tiện dụng, độ tinh vi, tính tin cậy, tính dễ bảo trì. Công nghệ: CSDL quan hệ, vi mạch siêu tích hợp, internet, mạng không dây tốc độ cao, hướng đối tượng, Web

Ngày nay phần mềm ngày càng lớn, càng tích hợp. Phần mềm ngày nay còn được truy cập ở khắp nơi trên thế giới thông qua Internet.

Người ta cho rằng việc xây dựng một phần mềm cũng cần tuân theo những nguyên tắc và kỷ luật giống như khi xây dựng một chiếc cầu hay thực hiện những công việc kỹ nghệ khác. Nếu một chiếc cầu bị lỗi khi xây dựng và do đó bị sập thì tác hại gây ra rất lớn. Một phần mềm quan trọng như phần mềm điều khiển hệ thống tên lửa đạn đạo của một nước mà có lỗi, cho kết quả tính toán sai thì hậu quả nó gây ra cũng thật là kinh khủng. Chính vì vậy một nhóm nghiên cứu của NATO trong **năm 1967** đã sử dụng thuật ngữ "**Software engineering**". Họ muốn rằng khi xây dựng phần mềm thì các kỹ sư cũng cần áp dụng các nguyên tắc của kỹ nghệ truyền thống.

Tuy nhiên, có nhiều sự khác biệt giữa sản phẩm của kỹ nghệ truyền thống và công nghệ phần mềm. Ví dụ như chiếc cầu và hệ điều hành chẳng hạn. Sự cố cầu sập rất ít khi xảy ra và nếu xảy ra thì cách khắc phục thường là xây dựng lại. Còn hệ điều hành nếu có sự cố có khi chỉ cần tắt máy khởi động lại là lại chạy tốt. Phần mềm có thể sửa lại, có khi là tới 50% để có thể chạy trên máy tính có cấu hình khác hoặc thực hiện thêm các chức năng khác. Còn chiếc cầu muốn sửa lại 50% thì rất khó, nhiều khi xây dựng mới còn dễ thực hiện hơn.

### **1.1.2 Từ góc độ kinh tế**

Trong kỹ nghệ truyền thống, khi có nhiều cách thức để sản xuất một sản phẩm nào đó, ví dụ chiết xuất dầu lửa từ than đá chẳng hạn, các kỹ sư thường chọn phương án rẻ nhất. Khi xây dựng phần mềm cũng vậy, người ta thường lựa chọn cách thức chi phí ít nhất. Một phương pháp lập trình mới có thể viết chương trình nhanh hơn, nhưng chi phí huấn luyện sử dụng và công bảo trì có thể lớn hơn khiến người ta phải cân nhắc khi lựa chọn.

### **1.1.3 Khủng hoảng phần mềm**

- Phần mềm được viết ngay từ khi xuất hiện các hệ máy tính và ngôn ngữ lập trình đầu tiên.

- Công nghệ phần cứng phát triển nhanh và mạnh. Tính năng và tiềm lực của nó ngày càng tăng. Các ứng dụng dựa trên phần cứng cũng phải phát triển theo, các phần mềm ngày càng trở nên phức tạp và khó hiểu.

- Trên thực tế nhu cầu ứng dụng công nghệ thông tin ngày càng cao. Mọi ngành nghề kinh tế, mọi hoạt động xã hội và đời sống kinh tế đều cần đến các phần mềm hỗ trợ. Dẫn đến các phần mềm ngày càng lớn và phức tạp.

=> Trên thực tế các sản phẩm phần mềm không đáp ứng kịp các yêu cầu của người sử dụng. Việc sản xuất phần mềm gặp thất bại quá nhiều. Hầu hết các sản phẩm phần mềm đều rơi vào tình trạng:

- + Không đáp ứng kịp các nhu cầu của người sử dụng
- + Vượt quá chi phí và thời hạn.
- + Tiềm ẩn nhiều lỗi trong các sản phẩm phần mềm
- + Không đảm bảo chất lượng.

Các dữ liệu quan sát được cho thấy:

- Cứ có 6 đề án được triển khai thì có 2 đề án bị thất bại
- 35% số dự án phần mềm thất bại vì các lý do: thời hạn, chi phí, chất lượng (không đáp ứng được nghiệp vụ, khó sử dụng, không tin cậy...)
- 45% : đã được phân phối, không được sử dụng
- 27% : không được phân phối
- 17% : bị hủy bỏ
- 6% : được sử dụng sau khi đã sửa đổi
- 5% : được sử dụng ngay sau khi phân phối



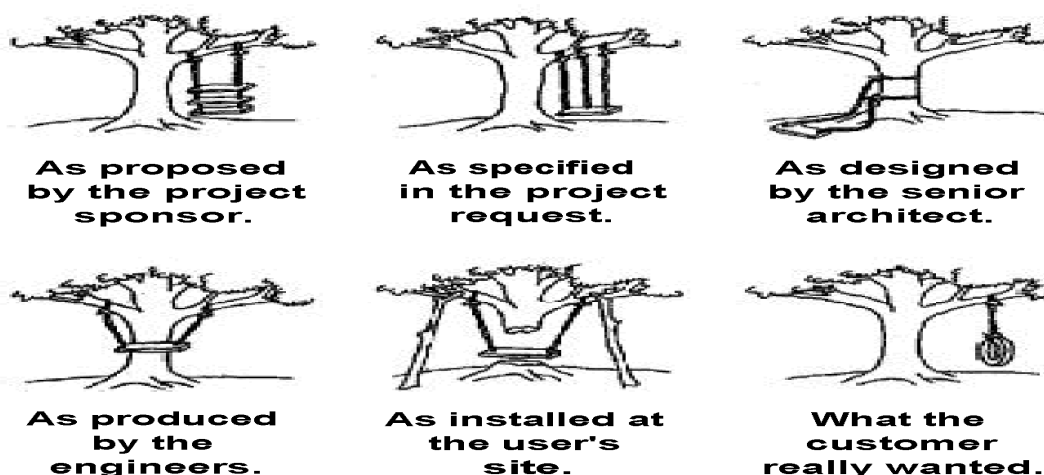
- Trung bình thời gian thực hiện thực tế bị kéo dài 50% (cá biệt lên tới 200 – 300%)
- 3/4 các hệ thống lớn có lỗi khi thực thi
- Quá trình phân tích yêu cầu (5% công sức): để lại 55% lỗi, có 18% phát hiện được
- Quá trình thiết kế (25% công sức): để lại 30% lỗi, có 10% phát hiện được
- Quá trình mã hóa, kiểm tra và bảo trì : Để lại 15% lỗi, có 72% phát hiện được

Để giải quyết vấn đề trên một hội nghị thế giới đã được triệu tập để bàn về cách giải quyết – Hội nghị của NATO về khủng hoảng phần mềm diễn ra 10/1968, đã:

Xác định các nguyên nhân chính gây ra khủng hoảng phần mềm, đó là việc sản xuất các sản phẩm phần mềm theo phương pháp thủ công. Phương pháp này không thích hợp cho việc phát triển các sản phẩm phần mềm lớn và phức tạp. Phương pháp thủ công thể hiện như sau:

- Làm theo cảm tính: Dựa chủ yếu vào kinh nghiệm, không có phương pháp đủ tốt
- Phương tiện thô sơ: Chủ yếu là ngôn ngữ lập trình
- Làm đơn lẻ: Do một hoặc một số cá nhân thực hiện

#### **Ví dụ Khủng hoảng phần mềm:**



#### **Khắc phục khủng hoảng phần mềm:**

Xây dựng phần mềm theo công nghệ ~ công nghiệp hóa quá trình sản xuất phần mềm. Từ đó Khái niệm công nghệ phần mềm được đưa ra. Và từ đó công nghệ phần mềm thực sự trở thành một ngành nghiên cứu không thể thiếu được trong lĩnh vực CNTT.

Để đáp ứng đòi hỏi của phát triển phần mềm cần có lý thuyết, kỹ thuật, phương pháp, công cụ đủ tốt để điều khiển tiến trình phát triển hệ thống phần mềm. Công nghệ phần mềm nhằm nghiên cứu tất cả các khía cạnh liên quan đến việc sản xuất các sản phẩm phần mềm chuyên nghiệp. Nó liên quan tới lý thuyết, quy trình, phương pháp và công cụ cần để phát triển phần mềm.

**Mục tiêu của công nghệ phần mềm:** Sản xuất phần mềm độc lập, đúng hạn, phù hợp kinh phí và đáp ứng mọi yêu cầu người sử dụng.

**Để xây dựng hệ thống phần mềm tốt ta cần**

- Xác định đúng đắn tiến trình phát triển phần mềm
  - Các pha của hoạt động
  - Sản phẩm của mỗi pha
- Phương pháp và kỹ thuật áp dụng trong từng pha và mô hình hóa sản phẩm của chúng
- Công cụ phát sinh ra sản phẩm

Các sản phẩm phần mềm ngày nay đã đóng góp một phần lớn cho nền kinh tế của đất nước và đạt được các kết quả nhất định. Mặc dù CNPM đã qua gần 2 thập kỷ phát triển, đã đưa ra nhiều nguyên lý, công cụ, phương pháp tiến trình.... để phát triển các sản phẩm phần mềm. Tuy nhiên các dự án phát triển phần mềm vẫn tồn tại chi phí cao, sản phẩm phần mềm chưa đảm bảo chất lượng và còn tiềm ẩn nhiều lỗi, các chi phí phần lớn vẫn dành cho hoạt động kiểm thử đánh giá phần mềm. Một số nguyên nhân (Những khó khăn trong phát triển phần mềm):

- Năng lực máy tính ngày càng mạnh
- Các hệ thống được liên kết lại ngày càng lớn
- Thế giới thay đổi nhanh (cả nghiệp vụ lẫn công nghệ)
- Ham muốn của người dùng ngày càng nhiều.

=> Yêu cầu tiến hóa phần mềm là tất yếu

- Quá trình phát triển phần mềm chưa được thống nhất
- Chưa đạt được chuẩn cho việc đo lường hiệu suất và sản phẩm
- Độ phức tạp phần mềm quá cao.
- Làm việc nhóm không đúng kỷ luật gây ra các lỗi
- Không có phương pháp mô tả rõ ràng định nghĩa yêu cầu của người dùng