

CHƯƠNG 1.

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

GVGD: Nguyễn Xuân Hà Giang

NỘI DUNG

- ▶ Giới thiệu tổng quan về Bảo trì phần mềm (BTPM)
- ▶ Tổ chức khung hoạt động của BTPM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Phần mềm (software): Theo McDermid phần mềm gồm
 - Các thuật ngữ

| | |
|-------------------------|---|
| <i>Chương trình</i> | <ol style="list-style-type: none"> 1. <i>Mã nguồn</i> 2. <i>Mã đối tượng</i> |
| <i>Tài liệu</i> | <ol style="list-style-type: none"> 1. <i>Đặc tả</i> 2. <i>Phân tích, thiết kế</i> 3. <i>Mã nguồn</i> 4. <i>Kiểm thử</i> |
| <i>Thủ tục vận hành</i> | <ol style="list-style-type: none"> 1. <i>Hướng dẫn cài đặt và sử dụng phần mềm</i> 2. <i>Hướng dẫn xử lý khi hệ thống gặp sự cố</i> |

► Các thuật ngữ

- Sự tiến hóa phần mềm (Software evolution): quy trình chuyển đổi từ trạng thái thấp, đơn giản, tệ hơn sang trạng thái cao, phức tạp, tốt hơn.
- 8 luật tiến hóa trong quy trình phần mềm của Lehman

| | |
|--|--|
| Luật về sự thay đổi liên tục (<i>Continuing change</i>) | Luật bảo toàn độ quen thuộc (<i>Conservation of familiarity</i>) |
| Luật về độ phức tạp luôn tăng (<i>Increasing complexity</i>) | Luật về sự phát triển liên tục (<i>Continuing growth</i>) |
| Luật về tự hiệu chỉnh (<i>Self-regulation</i>) | Luật về chất lượng xuống cấp (<i>Declining quality</i>) |
| Luật bảo toàn sự ổn định trong tổ chức (<i>Conservation of organisational stability</i>) | Luật về hệ thống phản hồi thông tin (<i>Feedback system</i>) |

- Bảo trì phần mềm (Software maintenance):
 - Maintenance: các hoạt động giữ cho thực thể ở trạng thái hiệu quả, hợp lệ, không bị sai hay xuống cấp
 - Các định nghĩa về BTPM:
 - Theo chuẩn 1219: mô tả quy trình quản lý và thực hiện các hoạt động BTPM của IEEE
 - ⇒ BTPM: việc sửa đổi sản phẩm phần mềm sau phát hành nhằm chỉnh lỗi, cải thiện khả năng thực thi hoặc tạo độ tương thích cho hệ thống và môi trường

- Theo chuẩn 12207: chuẩn cung cấp khung làm việc chung để phát triển và quản lý phần mềm của ISO/IEC
 - => BTPM: Sửa đổi mã nguồn, tài liệu phần mềm có liên quan đến vấn đề hay nhu cầu cần cải tiến trong khi phần mềm vẫn đảm bảo tính toàn vẹn
- Theo chuẩn 14764: chuẩn hợp nhất giữa 1219&12207 của ISO/IEC/IEEE
 - => BTPM: các hoạt động cần thiết để cung cấp sự hỗ trợ hiệu quả về chi phí cho phần mềm. Hoạt động này trải dài suốt quá trình phát triển, sau phát hành và vận hành sản phẩm.

- Các định nghĩa về BTPM:
- Theo chuẩn 14764 của ISO/IEC/IEEE

Các hoạt động phát triển phần mềm:

- Lập kế hoạch vận hành PM sau phát hành
- Duy trì, nâng cao tính có thể bảo trì
- Đưa ra các quyết định cho hoạt động chuyển giao,...

Hoạt động sau phát hành:

1. Hiệu chỉnh PM
2. Tập huấn nhân sự
3. Triển khai vận hành,...

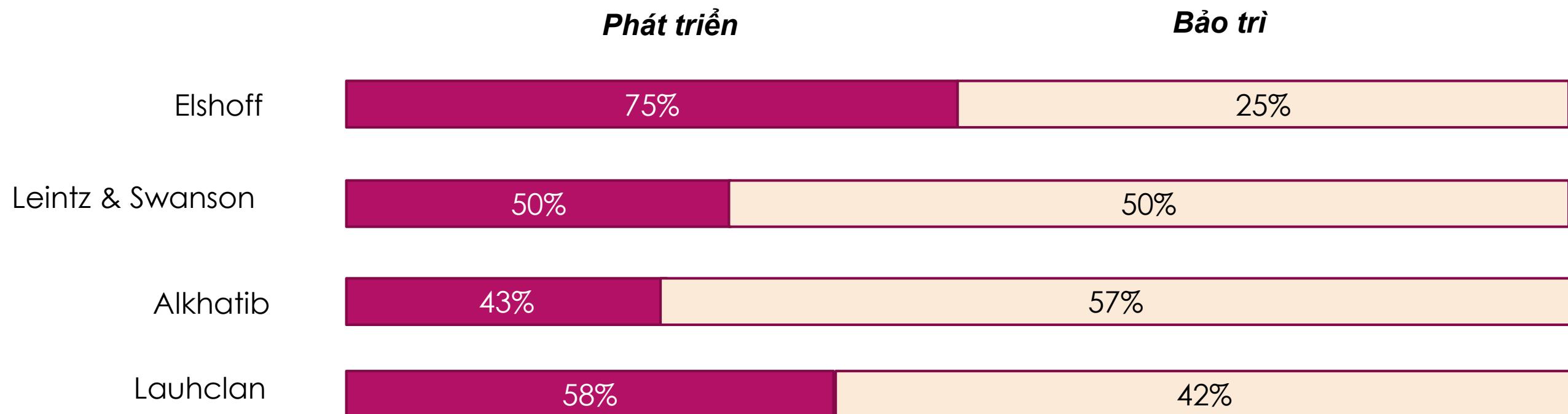
► Phân biệt BTPM & phát triển PM

| Phát triển phần mềm | Bảo trì phần mềm |
|--|--|
| <p>1. Phát triển mới</p> <p>2. Các ràng buộc xác định, rõ ràng</p> <p>3. Kiểm soát dựa trên “nhìn lại” giai đoạn trước</p> <p>4. Mục tiêu: tạo ra sản phẩm đáp ứng chính xác yêu cầu. Mỗi giai đoạn sau đều “nhìn lại” giai đoạn trước để định hướng</p> | <p>1. Sự tiếp tục của phát triển mới</p> <p>2. Các ràng buộc, thông số dựa trên hệ sống hiện hành</p> <p>3. Kiểm soát dựa trên “nhìn lại” sản phẩm hiện tại và “nhìn tới”</p> <p>4. Mục tiêu: phát triển mới sản phẩm hiện tại và dự tính đến các chức năng có thể thay đổi do nhu cầu như phần cứng, phần mềm giao diện</p> |

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Bảng thống kê chi phí bảo trì và phát triển phần mềm của các tác giả.
Chi phí BTPM chiếm từ 49% - 75%

► *Phân biệt BTPM & phát triển PM*



GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Khái niệm này xuất hiện cuối XX & phát triển mạnh từ 1997–2000
 - Cung cấp tính liên tục của dịch vụ
 - Hỗ trợ các nâng cấp bắt buộc
 - Hỗ trợ yêu cầu cần cải tiến cho NSD
 - Tạo tiền đề thuận tiện cho công tác BT tương lai
- *Phân biệt BTPM & phát triển PM*

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Sự cần thiết của BTPM: minh họa
 - 01/01/1999, sự kiện đồng tiền chung Châu Âu, 10% phần mềm toàn thế giới, 50% phần mềm khối liên minh Châu Âu cần bảo trì
 - 2000, sự cố máy tính Y2K, 75% các phần mềm trên thế giới cần bảo trì, kể cả các thiết bị nhúng trong y học, điện thoại, giếng dầu,...
 - 2038, sự kiện lịch của hệ điều hành UNIX hết hạn
 - 2050, cần thêm 1 số vào số an sinh xã hội Mỹ

► *Phân biệt BTPM & phát triển PM*

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Cung cấp tính liên tục của dịch vụ:
 - ▶ Một số hệ thống đòi hỏi hoạt động liên tục dù có phát hiện lỗi.
 - ▶ Ví dụ: điều khiển máy bay, tàu hỏa,... lỗi xảy ra hoặc dừng hệ thống đều đưa đến hệ quả xấu đến đời sống
- ▶ Hỗ trợ nâng cấp bắt buộc: nhu cầu đi với điều kiện bắt buộc trong thực tế
 - ▶ Thay đổi của chính phủ: luật thuế, luật thương mại
 - ▶ Để tăng tính cạnh tranh với thị trường: an toàn rút tiền ATM, tăng chất lượng dịch vụ
 - ▶ *Phân biệt BTPM & phát triển PM*

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Hỗ trợ yêu cầu cần cải tiến cho NSD:
 - Yêu cầu hệ thống tốt => nhiều NSD=> xuất hiện nhiều yêu cầu cải tiến, nâng cấp chức năng để hệ thống được tốt hơn
 - Yêu cầu cải thiện hiệu năng: tốc độ, khả năng tùy biến các mô hình làm việc,....
 - Tạo tiền đề thuận tiện cho công tác bảo trì tương lai: tái cấu trúc CSDL, chương trình, cập nhật tài liệu

► *Sự cần thiết của BTPM*

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- *Hỏi: theo A/C, việc BTPM, có luôn bắt nguồn từ các yêu cầu cần thay đổi hay không?*
- Một số loại yêu cầu về thay đổi phần mềm
 - Thay đổi hiệu chỉnh lỗi (corrective change): do hạn chế, xuất hiện lỗi trong PM. Lỗi có thể do thiết kế, luận lý hoặc lỗi lập trình

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- Thay đổi hiệu chỉnh lỗi (corrective change):
 - Lỗi thiết kế: phần mềm hoạt động không chính xác do thiết kế sai, truyền đạt nhầm ý
 - Lỗi luận lý: luồng lập luận thiếu cơ sở, bị lỗi, kiểm thử không đầy đủ dữ liệu cần thiết, hay việc cài đặt không chính xác với thiết kế
 - Lỗi lập trình: lỗi xử lý dữ liệu, truy xuất hệ thống hoặc thực hiện không chính xác với thiết kế luận lý

► Một số loại YCTĐ PM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi hiệu chỉnh lỗi (corrective change):
 - ▶ Lỗi thiết kế: phần mềm hoạt động không chính xác do thiết kế sai, truyền đạt nhầm ý
 - ▶ Lỗi luận lý: luồng lập luận thiếu cơ sở, bị lỗi, kiểm thử không đầy đủ dữ liệu cần thiết, hay việc cài đặt không chính xác với thiết kế
 - ▶ Lỗi lập trình: lỗi xử lý dữ liệu, truy xuất hệ thống hoặc thực hiện không chính xác với thiết kế luận lý
 - ⇒ Corrective change: còn được gọi là bug, làm cho PM không thỏa mãn đặc tả
- ▶ Một số loại YCTĐ PM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi hiệu chỉnh lỗi (corrective change):
 - ▶ Để xử lý bug đang làm ngừng hệ thống. Áp lực từ quản lý và NSD. Giải pháp “vá - patching” thường xuất hiện để sửa chữa khẩn cấp.
 - ▶ Hệ quả:
 - ⇒ Phát sinh nhiều vấn đề mới: phức tạp chương trình hơn, có khi gây hiệu ứng phụ tiềm ẩn cho các chức năng khác
 - ⇒ Làm thay đổi và suy thoái cấu trúc chương trình

▶ Một số loại YCTĐ PM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi thích ứng/ tích hợp (adaptive change):
 - ▶ Loại thay đổi phát sinh từ nhu cầu, phù hợp với môi trường của hoạt động của PM, chẳng hạn:
 - ▶ Môi trường ngoài tác động: thay đổi quy tắc, nghiệp vụ, chính sách, ...
 - ▶ Do sự di dời sang môi trường mới: phần mềm (trình biên dịch, hệ điều hành) hay phần cứng (bộ xử lý, thiết bị tích hợp,...)
 - ▶ *Một số loại YCTĐ PM*

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi hoàn thiện (perfective change):
 - ▶ Mở rộng những yêu cầu hiện tại của hệ thống: cải tiến chức năng, cải thiện hiệu quả tính toán. Mục tiêu:
 - ⇒ Hoàn thiện hệ thống góp phần tăng độ thành công, hữu ích của PM
 - ⇒ Thúc đẩy nhu cầu thay đổi dựa trên thực nghiệm, kinh nghiệm, tình huống thực tế của người dùng, mà ban đầu nó vượt ra khỏi, nằm ngoài phạm vi thiết kế

▶ Một số loại YCTĐ PM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi dự phòng (preventive change)
 - ▶ Ngăn ngừa khả năng sai lệch chức năng thiết kế trước đó; cải thiện khả năng bảo trì được của PM
 - ▶ Loại hình này phần lớn xuất phát từ maintainer, mục đích
 - ▶ Chương trình dễ hiểu hơn
 - ▶ Trong tương lai, dễ bảo trì hơn
 - ▶ Không làm thay đổi đáng kể về chức năng

▶ Một số loại YCTĐ PM

GIỚI THIỆU TỔNG QUAN VỀ BTPM

- ▶ Thay đổi dự phòng (preventive change)
 - ▶ Các hoạt động thay đổi dự phòng:
 - ▶ Kiến trúc lại mã lệnh: giảm độ phức tạp mã lệnh
 - ▶ Tối ưu hóa mã lệnh: hương trình chạy nhanh hơn, sử dụng bộ nhớ hiệu quả hơn
 - ▶ Cập nhật tài liệu: quản lý lịch sử chương trình, theo dõi các biến đổi PM sau mỗi lần thay đổi
 - ▶ *Một số loại YCTĐ PM*

- Mỗi quan hệ giữa các thay đổi PM:
 - Về nguyên tắc có thể phân biệt riêng lẻ các hoạt động bảo trì
 - Thực tế, hoạt động này đan xen nhau
 - ⇒ Để sửa đổi 1 PM chạy được trên hệ điều hành mới (thay đổi thích ứng) => các bug tiềm ẩn có thể xuất hiện (thay đổi hiệu chỉnh)

► Mỗi quan hệ giữa các thay đổi PM:

- Hiểu bản chất các thay đổi giúp đưa ra quyết định giải quyết nhanh hơn ở những loại cấp thiết
- Xếp độ ưu tiên để các thay đổi hiệu quả nhất



TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- Những yếu tố cản trở đưa đến quyết định thay đổi PM:
 - Hạn chế tài nguyên
 - Chất lượng hệ thống hiện hành
 - Chiến lược tổ chức
 - Sức ì của NSD
 - Khả năng thu hút và giữ chân nhân viên có năng lực
- *Thảo luận: “Sức ì của NSD” nên được giải quyết như thế nào?*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- Hạn chế tài nguyên: trở ngại chính về chất lượng và hiệu quả việc BTPM
 - Thiếu nhân viên có năng lực
 - Thiếu môi trường, công cụ phù hợp
 - Thiếu ngân sách
- *Những yếu tố cản trở sự thay đổi PM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- ▶ Chất lượng hệ thống hiện hành:
 - ▶ Hoạt động nhiều năm, nhiều lần nâng cấp, sửa chữa => các thay đổi có thể gây hiệu ứng lè khó dự đoán, nguy cơ sụp hệ thống
 - ▶ Chất lượng kém => khó thay đổi hệ thống => lỗi không được giải quyết => NSD tiếp tục với hệ thống lỗi thời, chi phí BT lớn

▶ *Những yếu tố cản trở sự thay đổi PM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- ▶ Chiến lược tổ chức: yếu tố cạnh tranh đưa đến tác động lớn trong việc quyết định ngân sách BT
- ▶ Sức ì của NSD: tâm lý NSD, cản trở sự thay đổi mặc dù tính chất có thể quan trọng và nhiều tiềm năng sinh lợi
- ▶ Sự thu hút và giữ nhân viên có năng lực: Chi phí bảo trì thường ở mức cao, một phần nằm vì thu hút và giữ chân maintainer có tài và kỹ năng tốt, để đảm bảo hệ thống hoạt động, đáp ứng người dùng

▶ *Những yếu tố cản trở sự thay đổi PM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- ▶ Bảo trì hiệu chỉnh, corrective maintenance
- ▶ Bảo trì thích ứng, adaptive maintenance
- ▶ Bảo trì hoàn thiện, perfective maintenance
- ▶ Bảo trì dự phòng, preventive maintenance
 - ⇒ Các loại này để đáp ứng tương ứng với các loại yêu cầu BT đã trình bày phần trên
 - ▶ *Các loại BTPM: theo ISO/IEC/IEEE 14764*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

► Bảo trì hiệu chỉnh: Khi lỗi xuất hiện, các maintainer quan tâm **nguyên nhân** đưa đến lỗi => chỉnh sửa, thay đổi yêu cầu, thiết kế, chương trình, công cụ kiểm thử và các tài liệu

⇒ Thông thường, các sửa chữa ban đầu tạm thời, đảm bảo hệ thống vẫn hoạt động được. Sau đó mới có kế hoạch cho việc sửa lỗi này

Hỏi: SV liên hệ ví dụ thực tế?

► *Các loại BTPM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

- Bảo trì thích ứng: thực hiện khi có YCTĐ phần cứng, phần mềm, nghiệp vụ hay chính sách nhà nước
 - ⇒ Thay đổi này giúp chương trình thích nghi được khi có 1 bộ phận cần thay đổi để tiến hóa
 - ⇒ Chẳng hạn, nâng cấp phiên bản CSDL, các chương trình cần bổ sung thông số để truy xuất

► *Các loại BTPM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

► Bảo trì hoàn thiện:

- Yêu cầu các maintainer thực hiện các thay đổi để cải tiến 1 phần nào đó của hệ thống (tài liệu, thiết kế, chương trình, kiểm thử, tìm kiếm sự cải tiến)
- Những thay đổi đó không bắt nguồn từ lỗi

► *Các loại BTPM*

TỔNG QUAN VỀ BẢO TRÌ PHẦN MỀM

► Bảo trì dự phòng: thay đổi 1 phần hệ thống để dự phòng các sự cố toàn hệ thống:

► Kiểm tra kiểu

► Bắt ngoại lệ

⇒ Lỗi thường được tìm thấy thực sự hoặc tiềm ẩn nhưng chưa gây ra thất bại hệ thống

► *Các loại BTPM*

KHUNG LÀM VIỆC BTPM

- ▶ Framework, khung làm việc gồm: tập khái niệm, điều kiện, giả thiết xác định cách thức hiểu, tiếp cận vấn đề
- ▶ Environment, môi trường: ảnh hưởng, điều kiện từ bên ngoài tác động làm ảnh hưởng đến khả năng hoạt động, hình dáng thực thể
 - ▶ Môi trường điều hành, operating environment
 - ▶ Môi trường tổ chức, organizational environment

▶ *Thuật ngữ*

► Môi trường:

- Môi trường điều hành: phần cứng, PM: hệ điều hành, trình biên dịch, web browser,... hoạt động đi theo sản phẩm PM
- Ví dụ: nâng cấp bộ xử lý => mã máy do trình biên dịch tạo cũng nâng cấp
- Môi trường tổ chức:

► Môi trường:

► Môi trường tổ chức: yếu tố không liên quan đến phần cứng, phần mềm. Ví dụ: chế độ, chính sách, quy định, quy tắc nghiệp vụ,...

⇒ Quy tắc nghiệp vụ, chính sách xuất hiện nhiều trong các hệ thống thông tin nên thường được tích hợp vào mã lệnh

⇒ *Thảo luận: liên hệ thực tế?*

► Nhân sự bảo trì:

- Quản lý hệ thống, chăm sóc khách hàng (helpdesk), đảm bảo chất lượng PM, tập huấn sử dụng
- Hiệu chỉnh, cải tiến, quản lý cấu hình PM, kiểm thử BT, hỗ trợ tài liệu, hỗ trợ tích hợp

- Quy trình bảo trì, maintenance process: mọi hoạt động thực hiện trong thời gian BT
- *Vấn đề về*: Nắm bắt các yêu cầu cần thay đổi
 - ⇒ Khó để nắm bắt tất cả yêu cầu, chỉ khi hệ thống đang đưa vào sử dụng, các yêu cầu mới thực sự rõ ràng
 - ⇒ NSD thiếu khả năng diễn đạt; phân tích viên chưa đủ nền tảng để nắm hết các YCTĐ => **lỗ hổng thông tin**

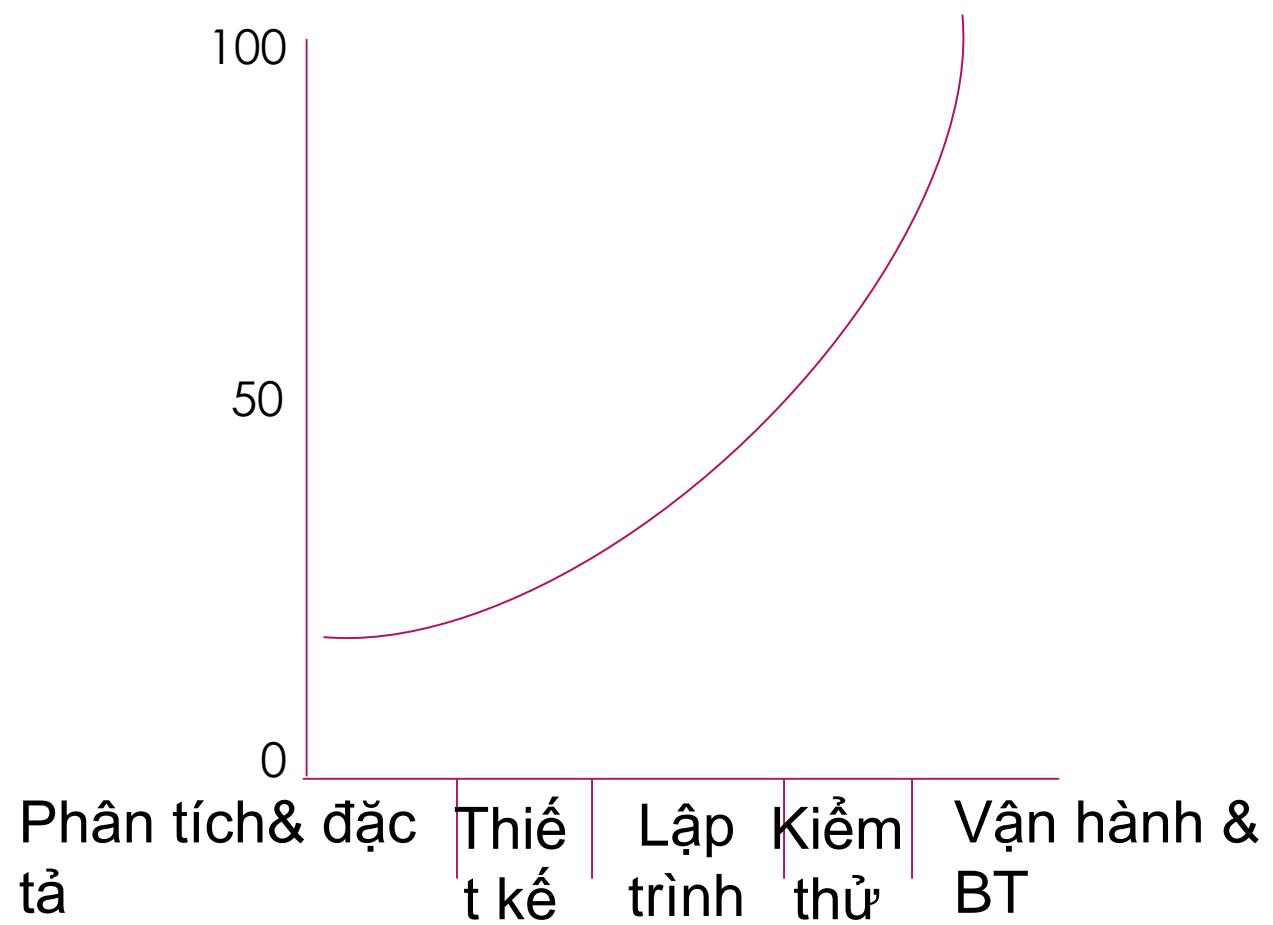
► Quy trình bảo trì, maintenance process:

- Thói quen lập trình khác nhau: trong cách tiếp cận, cách viết và cách BT của cá nhân hay tổ chức đưa đến sự không nhất quán
- Sự thay đổi trong cách thức phát triển và bảo trì PM
 - Cách lựa chọn ngôn ngữ lập trình để phát triển
 - Các hệ thống dùng ngôn ngữ cấp thấp trước khi ngôn ngữ cấu trúc, hướng đối tượng ra đời
 - Những PM hiện đang phát triển nhưng không được sử dụng công nghệ tiên tiến

- Quy trình bảo trì, maintenance process:
 - Sự thay đổi trong cách thức phát triển và bảo trì PM
 - Các PM này thừa hưởng các đặc trưng:
 - Dùng kỹ thuật, phương pháp không đủ để giao tiếp thiết bị; thiếu các đặc trưng: chương trình có cấu trúc, trừu tượng hóa dữ liệu & trừu tượng hóa chức năng

- Quy trình bảo trì, maintenance process:
 - Sự thay đổi trong cách thức phát triển và bảo trì PM
 - Các PM này thừa hưởng các đặc trưng:
 - Sử dụng ngôn ngữ và kỹ thuật lập trình không làm rõ cấu trúc & giao diện chương trình, kiểu & cấu dữ liệu
 - Không được quan tâm ràng buộc ảnh hưởng đến thiết kế
 - Mã lệnh thiết kế không theo chuẩn, chính thống, gây khó hiểu

- Quy trình bảo trì, maintenance process:
 - Khả năng phát hiện chỉnh sửa lỗi: yếu tố quan trọng của quy trình
 - Tồn tại thời điểm nhà sản xuất, người dùng biết họ cần gì và đồng ý với nhau về sản phẩm khi phát hành
 - Sau triển khai, lỗi được phát hiện hoặc không hài lòng chức năng
=> xuất hiện yêu cầu người sử dụng
 - ⇒ Lỗi càng phát hiện trễ, chi phí càng tăng, và rất cao khi vào giai đoạn BT



- Hiệu ứng gợn sóng, ripple effect: những ảnh hưởng của hoạt động tại nơi này lên nơi khác trong PM
- *Hỏi: SV liên hệ ví dụ?*

► Sản phẩm PM:

- Mã nguồn, mã đối tượng, tài liệu, thủ tục thiết lập, điều hành hệ thống. Là tất cả các vấn đề liên quan đến PM khi thay đổi
- Các mảng ảnh hưởng khó khăn đến hoạt động BT:

| | |
|--|--------------------------------------|
| Độ hoàn chỉnh của phân hệ chức năng, chương trình | Chất lượng tài liệu PM |
| Độ phức tạp của lĩnh vực ứng dụng | Tính dễ thay đổi chương trình |
| | Chất lượng kế thừa PM |

▶ Sản phẩm PM:

▶ Độ hoàn chỉnh của phân hệ chức năng, chương trình: các chức năng sử dụng phổ biến, rõ ràng, sẽ ít thay đổi hơn các yêu cầu chưa thành hình

⇒ Ví dụ: Hệ thống quản lý lương, quản lý nhân sự và hệ thống thông tin y khoa

KHUNG LÀM VIỆC BTPM

► Sản phẩm PM:

► Độ phức tạp của lĩnh vực ứng dụng:

► *Thuật ngữ*

⇒ Mô phỏng dịch rầy, dự báo thời tiết, biến động dân số

► Chất lượng tài liệu PM: vấn đề lớn thường đối mặt. Chương trình được sửa :

► Không được cập nhật tài liệu hoặc không chính xác

► Không có tài liệu nào

► Sản phẩm PM:

- Tính dễ thay đổi chương trình: khi phần mềm có đặc điểm này sẽ dễ nguy hiểm, gây ra hậu quả lớn khó lường trước, đặc biệt về bảo mật
- Chất lượng kế thừa: trạng thái tiến hóa phần mềm liên quan chặt chẽ đến trạng thái phần mềm liên đới

► Nhân sự bảo trì

- Đối tượng: quản lý viên, phân tích viên, thiết kế viên, lập trình viên, kiểm thử viên, bảo trì viên,...
- Chức năng: quản lý, bảo trì phần mềm, quản lý cấu hình, huấn luyện, kiểm thử, đảm bảo chất lượng, chăm sóc khách hàng, hỗ trợ tài liệu

► Nhân sự bảo trì:

- Quản lý viên: chịu trách nhiệm giám sát chung hoạt động bảo trì
- Bảo trì viên: trong bảo trì phần mềm
 - Sửa lỗi, tạo thích ứng phần mềm với môi trường thay đổi, cập nhật tính năng mới, cải tiến tính năng, phòng ngừa lỗi hoặc tạo môi trường dễ bảo trì hệ thống trong tương lai
 - Phân tích vấn đề, cô lập lỗi, phân tích yêu cầu cải tiến, thiết kế, lập trình, kiểm thử đơn vị, lập tài liệu thay đổi

KHUNG LÀM VIỆC BTPM

► Nhân sự bảo trì, chức năng:

► Bảo trì viên: quản lý cấu hình

► *Thuật ngữ*

► Cấu hình: các hoạt động quản lý sự thay đổi suốt thời gian sống của phần mềm

► Mục đích: đảm bảo phần mềm được nhận dạng, ghi lịch sử, triển khai phù hợp đặc tả và thiết kế.

KHUNG LÀM VIỆC BTPM

► Nhân sự bảo trì, chức năng:

► Bảo trì viên:

► *Thuật ngữ*

► Huấn luyện: Cung cấp thông tin, hướng dẫn người dùng vận hành chuỗi hoạt động trong phần mềm

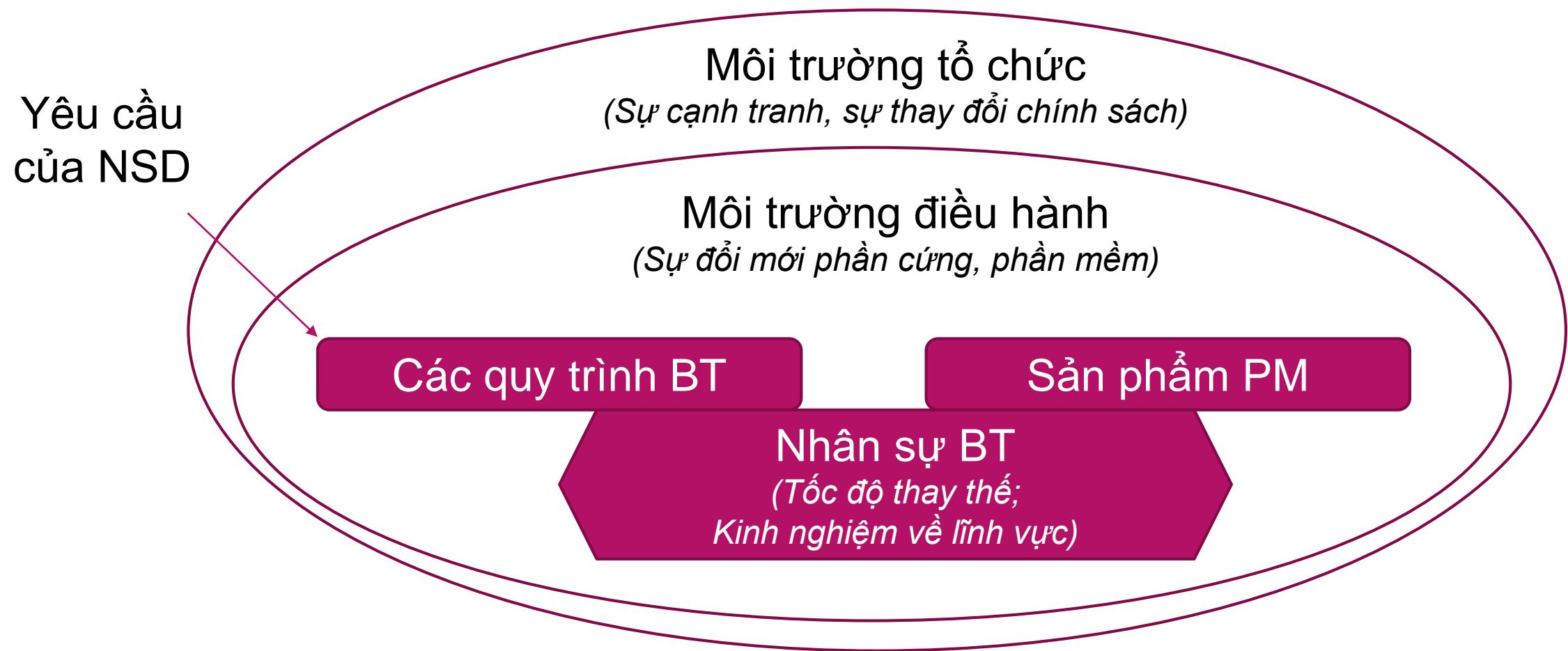
► Kiểm thử: thực hiện nhiều loại khác nhau (đơn vị, chức năng, tích hợp), lập kế hoạch kiểm thử và báo cáo kết quả

- Yếu tố ảnh hưởng công việc bảo trì:
 - Tốc độ thay nhân sự: phần lớn bảo trì viên không là tác giả ban đầu
 - ⇒ Khi quá trình thay đổi hoặc cố gắng thay đổi thành công, thất bại, giải thuật, giả thiết lựa chọn,... không được cập nhật tài liệu liên tục, bảo trì viên tiêu tốn nhiều thời gian và công sức để hiểu được đoạn mã cần thay đổi
 - ⇒ Hiểu chương trình tốn $\frac{1}{2}$ công sức bảo trì

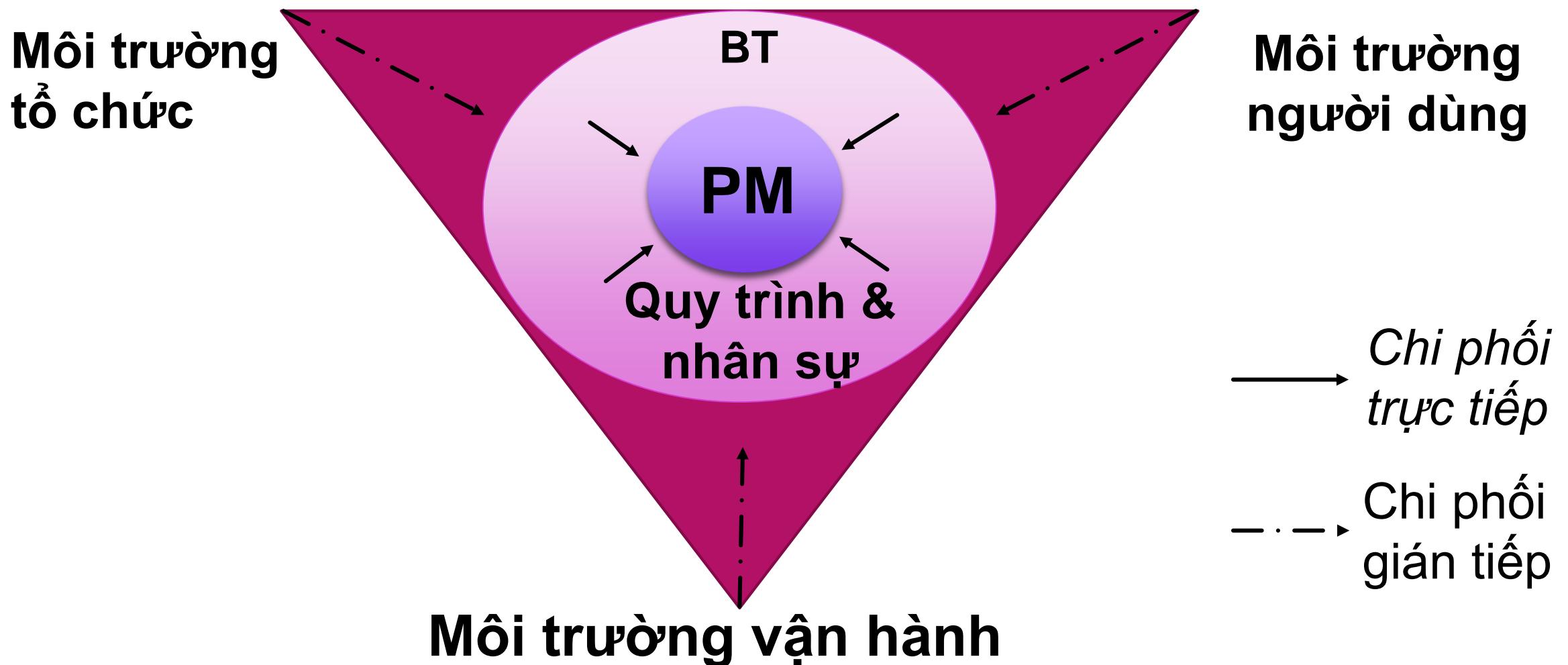
- ▶ Yếu tố ảnh hưởng công việc bảo trì:
 - ▶ Sự thành thạo về lĩnh vực chuyên môn hay hệ thống
 - ⇒ Khi bảo trì viên không được hỗ trợ (từ nhân viên chuyên môn hoặc tài liệu) thì hoạt động bảo trì có thể ảnh hưởng đến những thành phần khác của hệ thống (hiệu ứng gợn sóng) => chi phí bảo trì lớn
 - ▶ Kinh nghiệm thực tiễn: cách thức, giả thiết sử dụng, quyết định cài đặt, thiết kế của bảo trì viên tác động đến hệ thống, đưa đến mức độ dễ hiểu hay gây phức tạp hơn của kết quả

- NSD và các yêu cầu
 - Cá nhân sử dụng hệ thống, giai đoạn phát triển & BT
 - Các yêu cầu công việc BT:
 - Các yêu cầu chống lại sự suy thoái, cấu trúc PM tốt hơn, cập nhật tài liệu dễ hiểu và PM có khả năng phát triển thêm
 - Các yêu cầu về tiến hóa để tinh chỉnh chức năng hiện tại hoặc bổ sung mới

► Các thành phần trong framework



► *Mối quan hệ các thành phần trong framework*



KHUNG LÀM VIỆC BTPM

- ▶ Sản phẩm & môi trường
- ▶ Sản phẩm & NSD (người dùng)
- ▶ Sản phẩm & nhân sự BT

▶ *Mối quan hệ các thành phần trong framework*

KHUNG LÀM VIỆC BTPM

- Phản ánh thông qua sự thay đổi đặc điểm trong PM:
 - Kích thước chương trình: đo theo số module hay số dòng lệnh, thay đổi do môi trường hoạt động và theo nhu cầu NSD
⇒ Xu hướng tăng khi có sự tăng cường hệ thống
 - Độ phức tạp: đo theo mức độ khó của việc hiểu mã nguồn
=> Xu hướng tăng khi cần sửa đổi chương trình
 - *Mối quan hệ các thành phần trong framework*

► *Mối quan hệ các thành phần trong framework*

- Sản phẩm & môi trường:
 - ⇒ PM thuộc về cơ chế điều hành và của 1 tổ chức nhất định, nên sẽ kế thừa những thay đổi có liên quan đến chức năng, chính sách,...
- Sản phẩm & NSD:
 - ⇒ Sản phẩm luôn thay đổi theo nhu cầu
- Sản phẩm & nhân sự BT:
 - ⇒ Phương pháp thực hiện & khả năng nhân sự BT ảnh hưởng chất lượng PM

1. Nêu vài sự kiện ở Việt Nam và thế giới gây tác động và làm tăng tính cần thiết của hoạt động BT đồng thời thúc đẩy phát triển nhanh?
2. Nêu một số quy định được chính phủ Việt Nam điều chỉnh gần đây và chỉ ra các loại PM bị ảnh hưởng?
3. Nêu 1 sản phẩm PM bạn đang sử dụng, và một số hạn chế (chức năng, ràng buộc, ...), đưa ra những YCTĐ , phân loại chúng và giải thích việc phân loại đó?

- ▶ Nếu một số phiên bản & tên 1 sản phẩm PM bạn đang sử dụng, 1 số thay đổi trong phiên bản đó và phân loại thuộc nhóm thay đổi nào?
- ▶ Nếu lỗi hỏng thông tin xuất hiện khi bảo trì viên tiếp nhận yêu cầu thay đổi từ người dùng, hãy nêu các biện giảm giúp giảm vấn đề này?
- ▶ Các lập trình viên trong 1 dự án, thường có thói quen lập trình khác nhau, gây khó khăn BT. Nêu giải pháp cho vấn đề này?

| Hiện tại | Yêu cầu |
|---|--|
| <i>Khi xóa 1 chi đoàn, các đoàn viên không bị xóa khỏi CSDL</i> | <i>Tự động xóa các đoàn viên mỗi khi chi đoàn bị xóa khỏi CSDL</i> |
| <i>PM đang có HQT CSDL Ms Access, không phù hợp thực tiễn</i> | <i>Dùng hệ quản trị CSDL MySQL</i> |
| <i>Chưa cung cấp đầy đủ các hạng mục trong chức năng báo cáo</i> | <i>Cho cán bộ đoàn khoa in danh các chi đoàn trực thuộc tham gia sự kiện theo thời gian; in danh sách BCH đoàn khoa các nhiệm kỳ</i> |
| <i>PM chưa kiểm tra sự hợp lý giữa ngày sinh và ngày vào đoàn</i> | <i>Kiểm tra mỗi liên hệ ngày sinh và ngày vào đoàn</i> |
| <i>Tài liệu đặc tả chưa phù hợp hệ thống</i> | <i>Cập nhật tài liệu để phản ánh đúng hệ thống</i> |