

# CTT521 - Các Chủ Đề Nâng Cao Trong Công Nghệ Phần Mềm

## Bài tập về nhà 1

Đọc, nghiên cứu, và tóm tắt bài

Brooks, Frederick P., "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer*, Vol. 20, No. 4 (April 1987) pp. 10-19.

1. (20 điểm) Tóm tắt những điểm chính của bài báo, tối đa 1 trang

Một số từ ngữ cần lưu ý:

- **no silver bullet** có nghĩa bóng là **không có giải pháp hoàn hảo** làm cho giá thành phát triển phần mềm giảm như giá thành sản xuất phần cứng. Nên nhớ rằng theo định luật Moore, tốc độ xử lý của máy tính tăng gấp đôi sau mỗi 2 năm.
- **Essential difficulties** là những khó khăn cố hữu thuộc về bản chất của phần mềm
- **Accidental difficulties** là những khó khăn còn lại (không thuộc bản chất của phần mềm). Có thể dịch là khó khăn ngẫu nhiên, khó khăn phụ thuộc. Không thể dịch là tai nạn.

**Tóm tắt tốt cần phải nêu được các điểm chính sau:**

- Công nghệ phần cứng đã và đang phát triển vượt bậc làm giảm giá thành sản phẩm nhanh chóng. Vậy có các giải pháp mang tính đột phá như vậy trong công nghệ phần mềm hay không?
- Tác giả phân các khó khăn trong phát triển phần mềm thành hai loại: khó khăn thuộc về bản chất và các khó khăn còn lại.
- Khó khăn thuộc về bản chất xuất phát từ các thuộc tính cố hữu của phần mềm bao gồm Tính phức tạp (Complexity), Tính tương thích (Conformity), Tính thay đổi (Changeability), và Tính vô hình (Invisibility).
- Khó khăn trong phát triển phần mềm nằm ở đặc tả, thiết kế, và kiểm tra tính đúng đắn của các cấu trúc thuộc về khái niệm (conceptual construct) chứ không phải ở việc viết mã nguồn và kiểm tra mã nguồn.
- Các giải pháp đã đạt được như ngôn ngữ cấp cao, môi trường lập trình tích hợp, v.v..., chỉ là các giải pháp giải quyết các khó khăn không thuộc về bản chất của phần mềm.
- Những giải pháp khác được mong đợi là các giải pháp hoàn hảo giải quyết các khó khăn bản chất của phần mềm như ngôn ngữ Ada và các ngôn ngữ cấp cao, lập trình hướng đối tượng, v.v... **Tuy nhiên, tác giả cho rằng những giải pháp này cũng không phải là những giải pháp hoàn hảo nốt.**
- Tác giả đề nghị một số giải pháp hứa hẹn có thể giải quyết khó khăn về đặc tả, thiết kế, và kiểm tra tính đúng đắn của các cấu trúc thuộc về khái niệm. Các giải pháp như mua thay vì phát triển những thành phần (components), quy trình tinh chỉnh yêu cầu phần mềm và phát triển prototype, đào tạo và khuyến khích những người thiết kế giỏi.
- **Tóm lại: không có giải pháp hoàn hảo (no silver bullet)**

2. (15 điểm) Nêu và giải thích một ví dụ từ kinh nghiệm và hiểu biết của bản thân để xác nhận hay phản bác một trong các điểm mà tác giả của bài báo đã nêu.

Nêu các ví dụ như

- Cho dù có các công cụ lập trình tích hợp như Visual studio, Eclipse, v.v... nhưng thời gian của lập trình viên vẫn không được rút ngắn bao nhiêu bởi vì phần lớn thời gian họ bỏ ra để suy nghĩ và đưa ra cấu trúc mã nguồn và thực hiện cấu trúc đó.
- Thiết kế bao giờ cũng khó hơn là lập trình.
- Giá thành phần mềm không giảm với tốc độ như giá thành phần cứng.
- Hầu hết các phần mềm bây giờ sử dụng ít nhất một vài thành phần đã được phát triển đầu đó.
- V.v...

3. (15 điểm) Hãy thử tóm tắt bài báo trong tối không quá 500 từ.

Câu này yêu cầu viết ngắn gọn **các ý chính** trong câu 1 ở trên theo dạng abstract thường thấy ở các bài báo khoa học.

Sau đây một ví dụ:

Sự phát triển vượt bậc của công nghệ phần cứng làm giảm giá thành sản phẩm một cách nhanh chóng. Liệu có những giải pháp hoàn hảo nào trong công nghệ phần mềm làm tăng năng suất, tăng độ tin cậy và đơn giản của phần mềm với mức độ như phần cứng hay không? Tác giả phân tích những giải pháp đã đạt được trong công nghệ phần mềm và kết luận rằng những giải pháp này tuy mang lại một số thành quả nhất định nhưng không thể giải quyết những khó khăn thuộc về bản chất của việc phát triển phần mềm. Những khó khăn này xuất phát từ bản chất của phần mềm bao gồm Tính phức tạp, Tính tương thích, Tính thay đổi, và Tính vô hình. Vì vậy, tác giả kết luận rằng không có giải pháp hoàn hảo nhằm giảm giá thành trong công nghệ phần mềm. Mặc dù vậy, một số giải pháp hứa hẹn có thể giải quyết khó khăn thuộc về bản chất như mua thay vì phát triển, quy trình tinh chỉnh yêu cầu và phát triển nhanh prototype, đào tạo và khuyến khích những người thiết kế giỏi.