Kiểm tra Phần mềm

Bài 02.

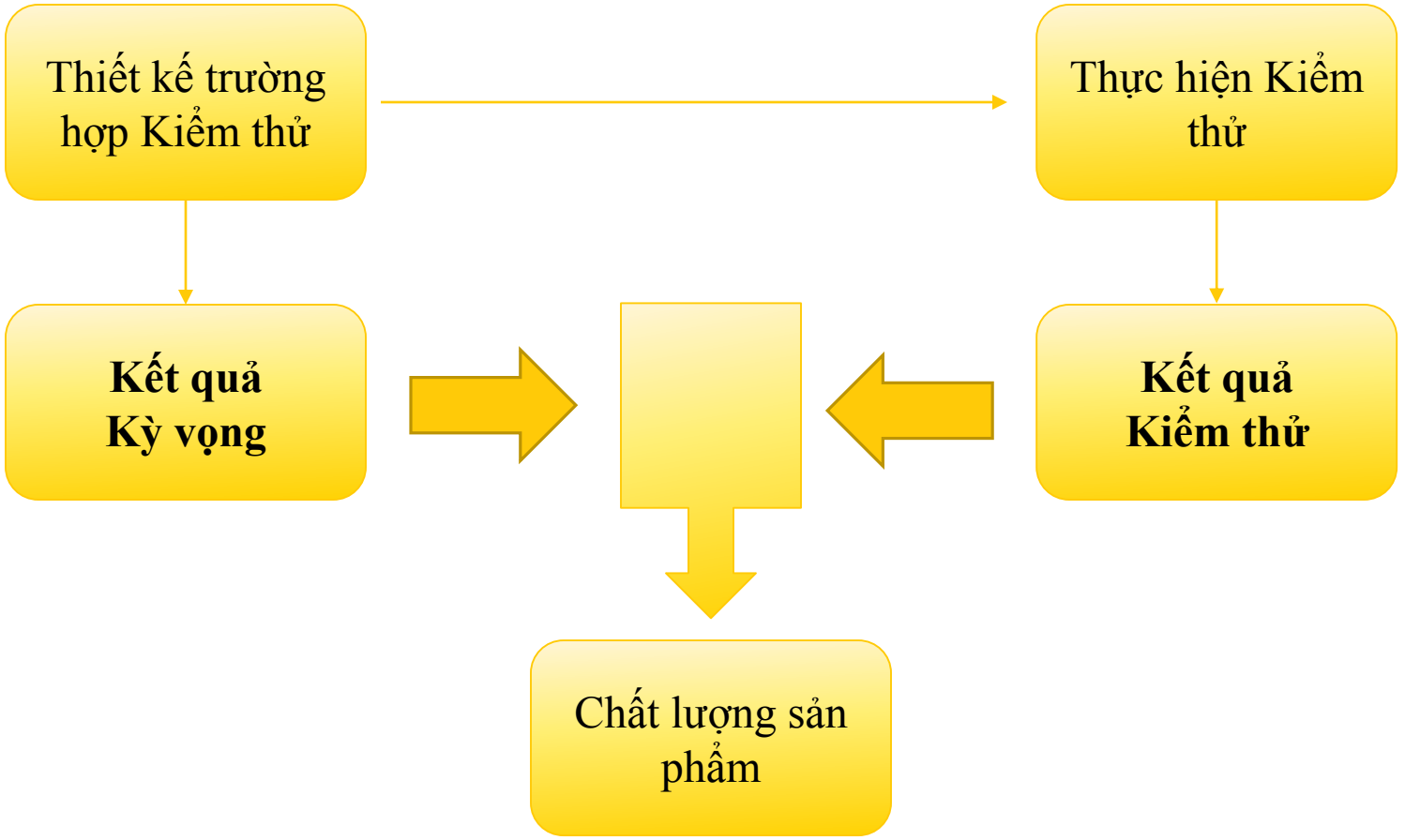
Các nguyên lý kiểm tra phần mềm

Lecturer: **Nguyễn Ngọc Tú**
Email: *Tu.NguyenNgoc@hoasen.edu.vn*
Web: *sites.google.com/site/WebKiemTraPhanMem/*
#AdTekDev #ICoTek

Nội dung

- Quy trình kiểm tra cơ bản
- Lập kế hoạch và kiểm soát
- Phân tích và thiết kế
- Hiện thực và việc thực thi
- Đánh giá các tiêu chuẩn và lập báo cáo
- Các hoạt động sau kiểm tra
- Triết lý của việc kiểm tra phần mềm

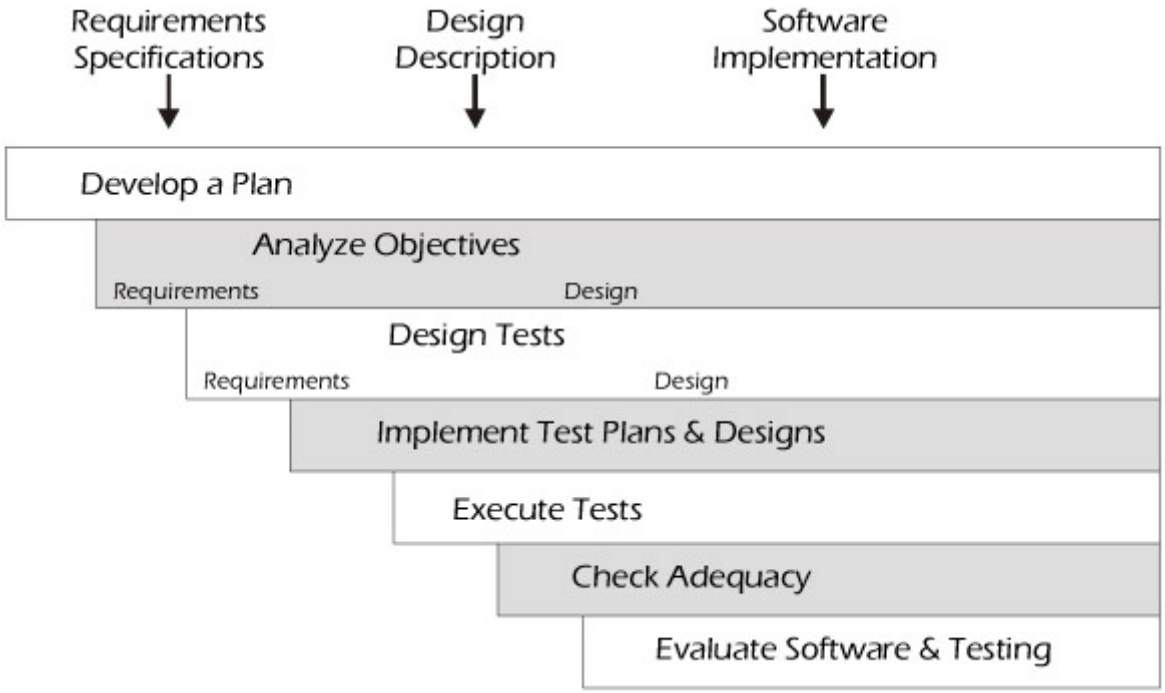
Quy trình kiểm tra cơ bản



Quy trình kiểm tra cơ bản

Quy trình kiểm tra cơ bản

- STEP *(Systematic Test and Evaluation Process)*



Systematic Software Testing
Rick D. Craig , Stefan P. Jaskiel

Lập kế hoạch và kiểm soát

- Xác định rõ
  - Phạm vi
  - Rủi ro
  - Mục tiêu kiểm thử
- Xác định cách tiếp cận
- Lập lịch thực hiện
- Đo lường và theo dõi tiến độ
- Điều chỉnh kế hoạch theo thời gian

Phân tích và thiết kế

- Rà soát lại điều kiện cơ sở để thực hiện kiểm thử
- Xác định các điều kiện kiểm thử
- Thiết kế trường hợp kiểm thử
- Đánh giá khả năng kiểm thử
- Xác định môi trường kiểm thử

Hiện thực và việc thực thi

- Phát triển và lập ưu tiên các trường hợp kiểm thử
- Tạo dữ liệu kiểm thử
- Viết đặc tả TC, thủ tục kiểm thử
- Viết mã thực thi kiểm thử
- Tạo nhóm kiểm thử theo đặc thù yêu cầu
- Kiểm soát môi trường kiểm thử theo như cấu hình

Đánh giá các tiêu chuẩn và lập báo cáo

- Kiểm tra, so sánh kết quả kiểm thử
- Đánh giá tiêu chuẩn chất lượng đạt được
- Viết báo cáo tổng hợp

Các hoạt động sau kiểm tra

- Tập hợp dữ liệu, mã kiểm thử
- Rút kinh nghiệm
- Cải tiến quy trình kiểm thử

Triết lý của việc kiểm tra phần mềm

1. Chỉ cho thấy lỗi xuất hiện
2. Không thể kiểm thử hết mọi trường hợp
3. Kiểm tra sớm
4. Lỗi xảy ra theo "cụm"
5. Nghịch lý "thuốc trừ sâu"
6. Phụ thuộc ngữ cảnh hệ thống
7. "Ảo tưởng" không lỗi

ST – Myths

- *Testing is Too Expensive*
- *Testing is Time-Consuming*
- *Only Fully Developed Products are Tested*
- *Complete Testing is Possible*
- *A Tested Software is Bug-Free*
- *Missed Defects are due to Testers*
- *Testers are Responsible for Quality of Product*
- *Test Automation should be used wherever possible to Reduce Time*
- *Anyone can Test a Software Application*
- *A Tester's only Task is to Find Bugs*

Chuẩn cần áp dụng

- **IEEE 829**
- **IEEE 1061**
- **ISO/IEC 9126**
- ISO/IEC 9241-11
- ISO/IEC 25000:2005
- ISO/IEC 250xx - **SQuaRE**
- ISO/IEC 12119
- IEEE 1059
- IEEE 1008
- IEEE 1012
- IEEE 1028
- IEEE 1044
- IEEE 1044-1
- **IEEE 830**
- **IEEE 730**
- IEEE 12207
- BS 7925-1
- BS 7925-2
- IEEE 610
- ISO 27000, ...

Q/A ?!

The Most Important Tests (MITs) Method

- What do we think we know about this project?
- How big is the test effort?
- If we can't test everything, what should we test?
- How long will the effort take?
- How much will it cost? (How much can we get?)
- How do I identify the tests to run?
- Are we on schedule? Have we tested enough?
- How successful was the test effort? Was the test coverage adequate? Was the test effort adequate?

Software Testing Fundamentals: Methods and Metrics
by Marnie L. Hutcheson

The Most Important Tests (MITs) Method – STEPs

1. State your assumptions.
2. Build the test inventory.
3. Perform MITs analysis.
4. Estimate the test effort.
5. Negotiate for the resources to conduct the test effort.
6. Build the test scripts.
7. Conduct testing and track test progress.
8. Measure test performance.

Software Testing Fundamentals: Methods and Metrics
by Marnie L. Hutcheson