BINUS UNIVERSITY

People
Innovation
Excellence

Course	: COMP6100/Software Engineering
Effective Period	: Desember 2017

Software Quality Assurance and Software Testing Strategies

Session 16

Acknowledgement

These slides have been adapted from Pressman, R.S. (2015). *Software Engineering : A Practioner's Approach. 8th ed*. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157. Chapter 21 and 22

People
Innovation
Excellence

Learning Objectives

LO 3 : Demonstrate the quality assurances and the potential showcase business project

Contents

Elements of SQA

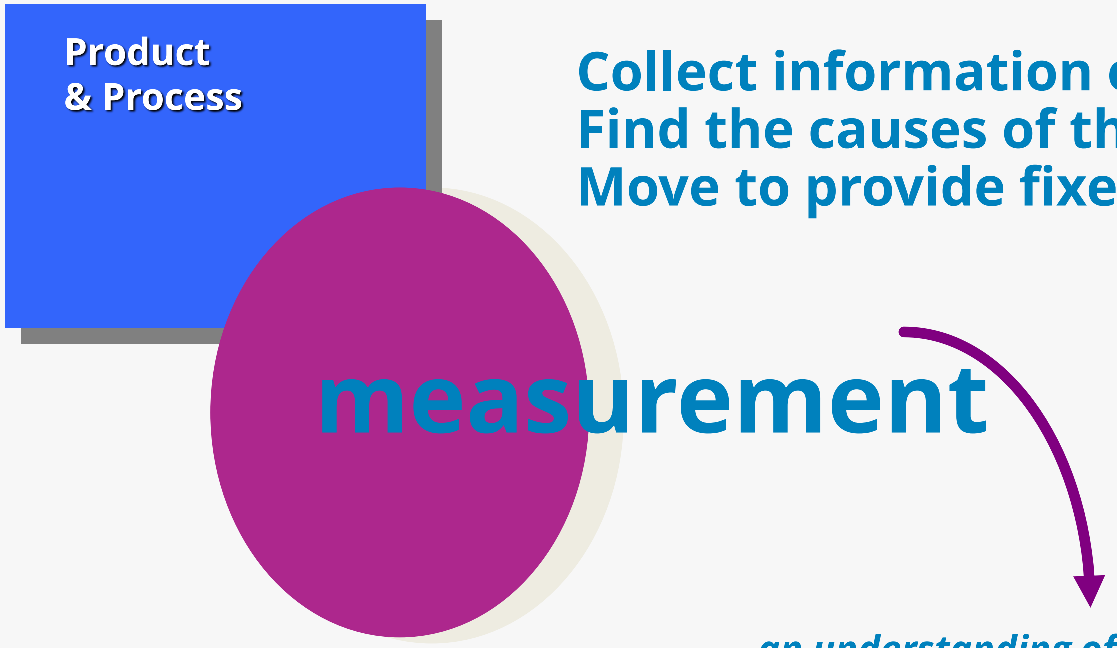
- Standards
- Reviews and Audits
- Testing
- Error/defect collection and analysis
- Change management
- Education
- Vendor management
- Security management
- Safety
- Risk management

SQA Goals

- **Requirements quality.** The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- **Design quality.** Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.
- **Code quality.** Source code and related work products (e.g., other descriptive information) must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- **Quality control effectiveness.** A software team should apply limited resources in a way that has the highest likelihood of achieving a high quality result.

People
Innovation
Excellence

Statistical SQA



Collect information on all defects
Find the causes of the defects
Move to provide fixes for the process

... an understanding of how

to improve quality ...

Statistical SQA

- Information about software errors and defects is collected and categorized.
- An attempt is made to trace each error and defect to its underlying cause (e.g., non-conformance to specifications, design error, violation of standards, poor communication with the customer).
- Using the Pareto principle (80 percent of the defects can be traced to 20 percent of all possible causes), isolate the 20 percent (the *vital few*).
- Once the vital few causes have been identified, move to correct the problems that have caused the errors and defects.

People
Innovation
Excellence

Six-Sigma for Software Engineering

- The term “six sigma” is derived from six standard deviations—3.4 instances (defects) per million occurrences—implying an extremely high quality standard.
- The Six Sigma methodology defines three core steps:
  - *Define* customer requirements and deliverables and project goals via well-defined methods of customer communication
  - *Measure* the existing process and its output to determine current quality performance (collect defect metrics)
  - *Analyze* defect metrics and determine the vital few causes.
  - *Improve* the process by eliminating the root causes of defects.
  - *Control* the process to ensure that future work does

Software Reliability

- A simple measure of reliability is *mean-time-between-failure* (MTBF), where

  MTBF = MTTF + MTTR

- The acronyms MTTF and MTTR are *mean-time-to-failure* and *mean-time-to-repair*, respectively.
- *Software availability* is the probability that a program is operating according to requirements at a given point in time and is defined as

  Availability = [MTTF/(MTTF + MTTR)] x 100%

Software Safety

- *Software safety* is a software quality assurance activity that focuses on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail.
- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.

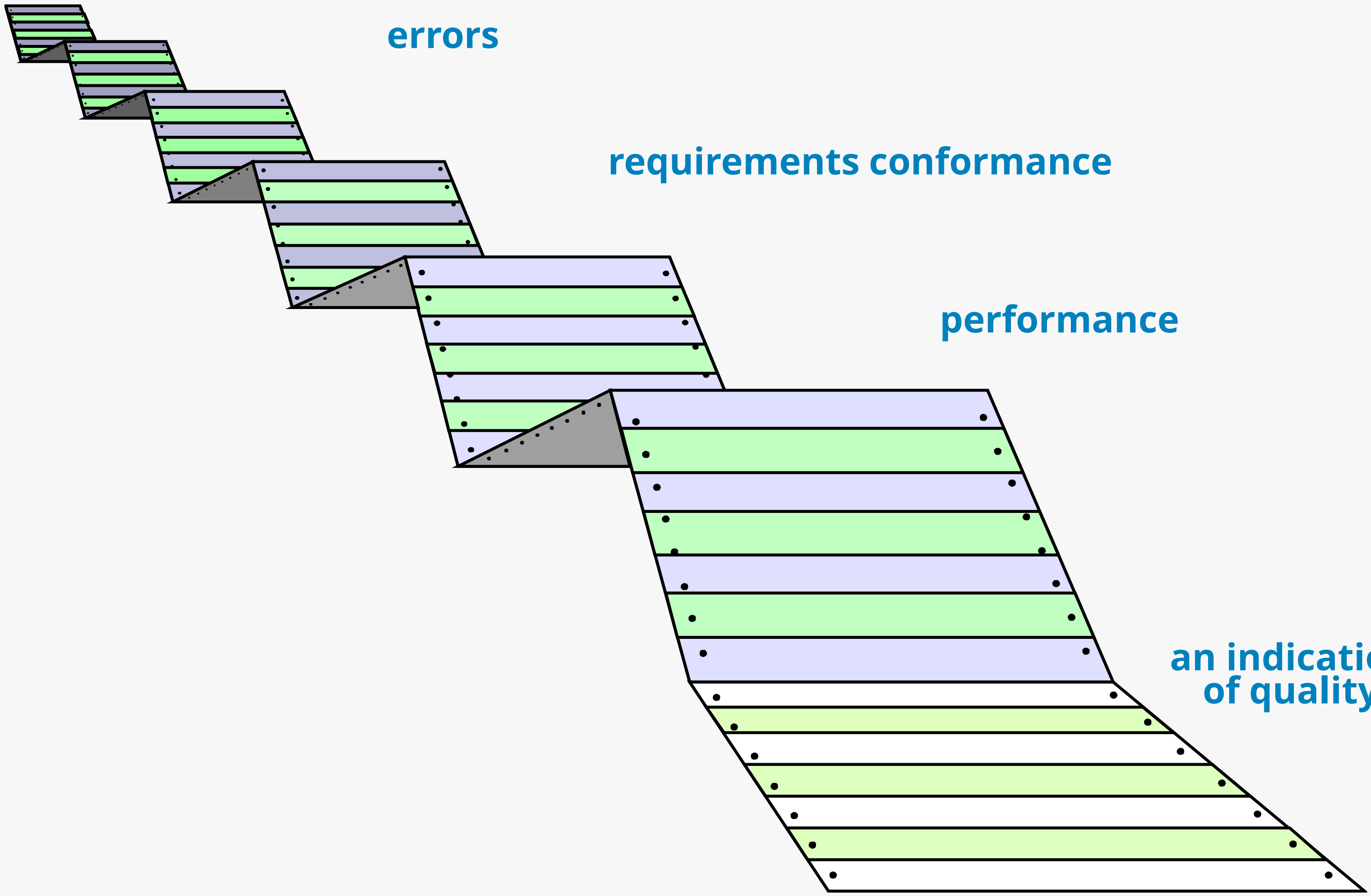
Software Testing

Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.

Software Testing

BINUS UNIVERSITY

What Testing Shows



People
Innovation
Excellence

Software Testing

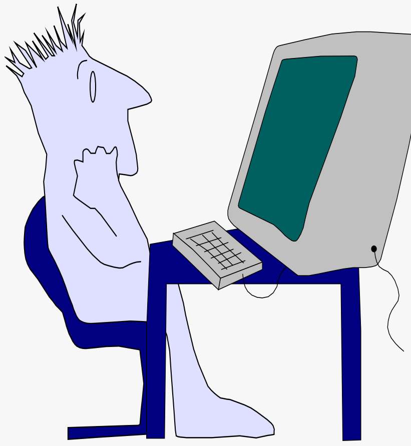
V & V

- *Verification* refers to the set of tasks that ensure that software correctly implements a specific function.
- *Validation* refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. Boehm [Boe81] states this another way:
  - *Verification:* "Are we building the product right?"
  - *Validation:* "Are we building the right product?"

People
Innovation
Excellence

Software Testing

Who Tests the Software?



developer

independent tester

Understands the system

but, will test "gently"

and, is driven by "delivery"

Must learn about the system,

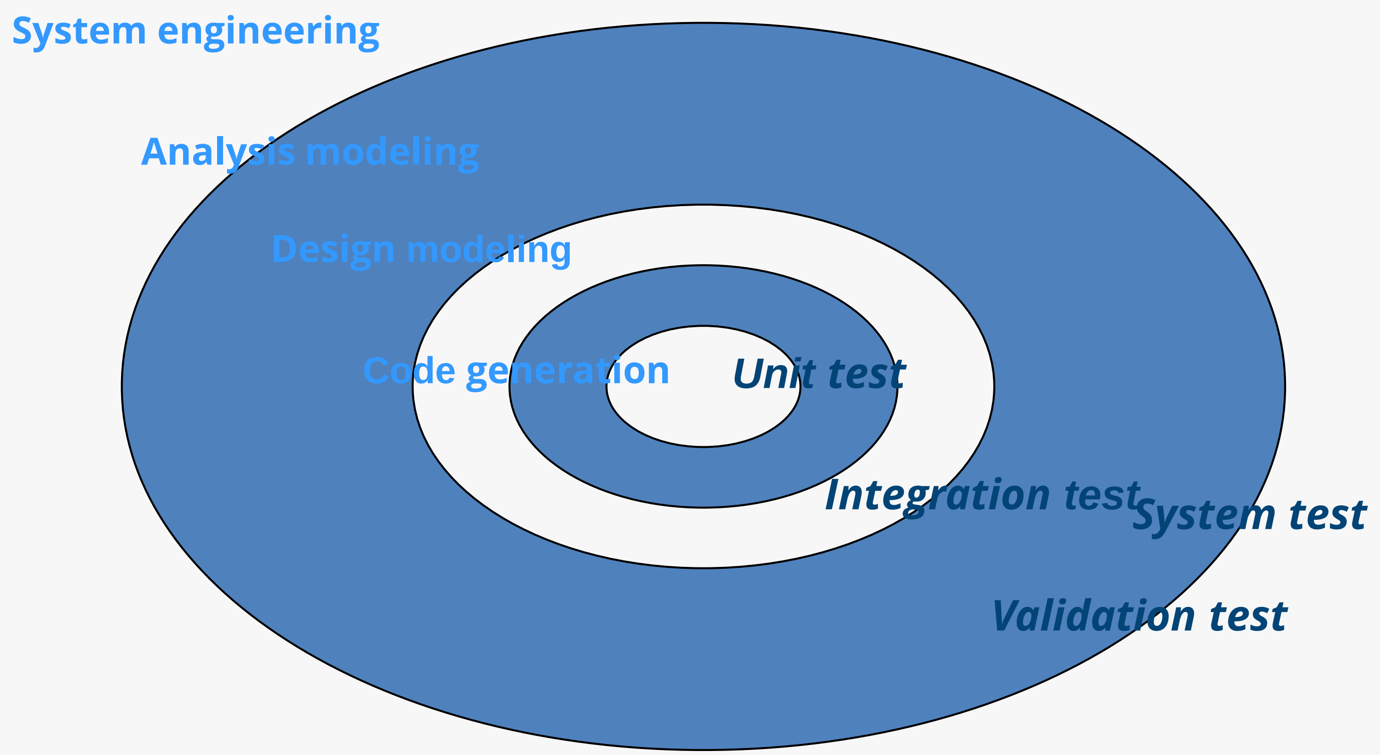
but, will attempt to break it

and, is driven by quality

People
Innovation
Excellence

People
Innovation
Excellence

Strategic Testing



Strategic Testing

Testing Strategy

- We begin by ‘testing-in-the-small’ and move toward ‘testing-in-the-large’
- For conventional software
  - The module (component) is our initial focus
  - Integration of modules follows
- For OO software
  - our focus when “testing in the small” changes from an individual module (the conventional view) to an OO class that encompasses attributes and operations and implies communication and collaboration

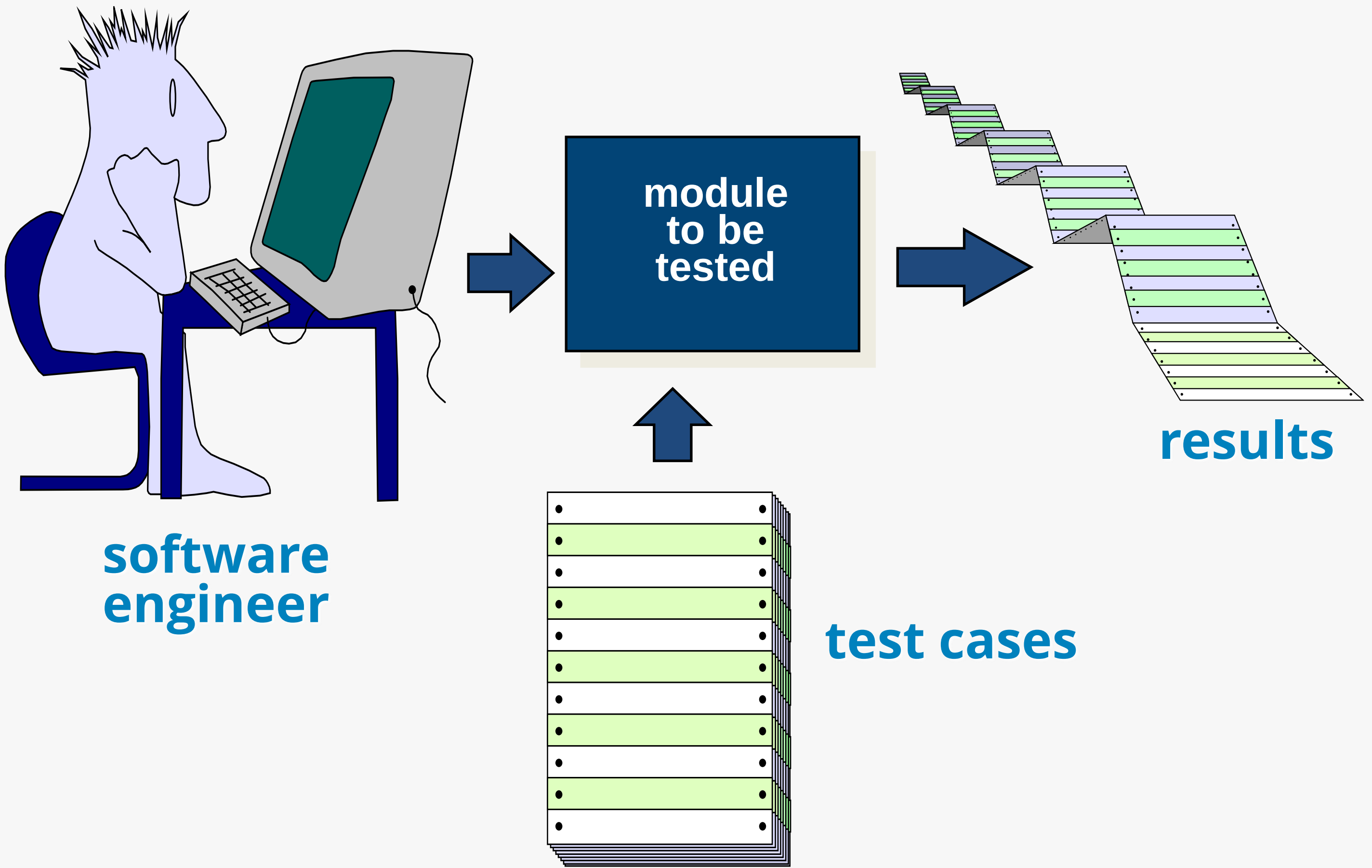
Strategic Testing

Strategic Issues

- Specify product requirements in a quantifiable manner long before testing commences.
- State testing objectives explicitly.
- Understand the users of the software and develop a profile for each user category.
- Develop a testing plan that emphasizes “rapid cycle testing.”
- Build “robust” software that is designed to test itself
- Use effective technical reviews as a filter prior to testing
- Conduct technical reviews to assess the test strategy and test cases themselves.
- Develop a continuous improvement approach for the testing process.

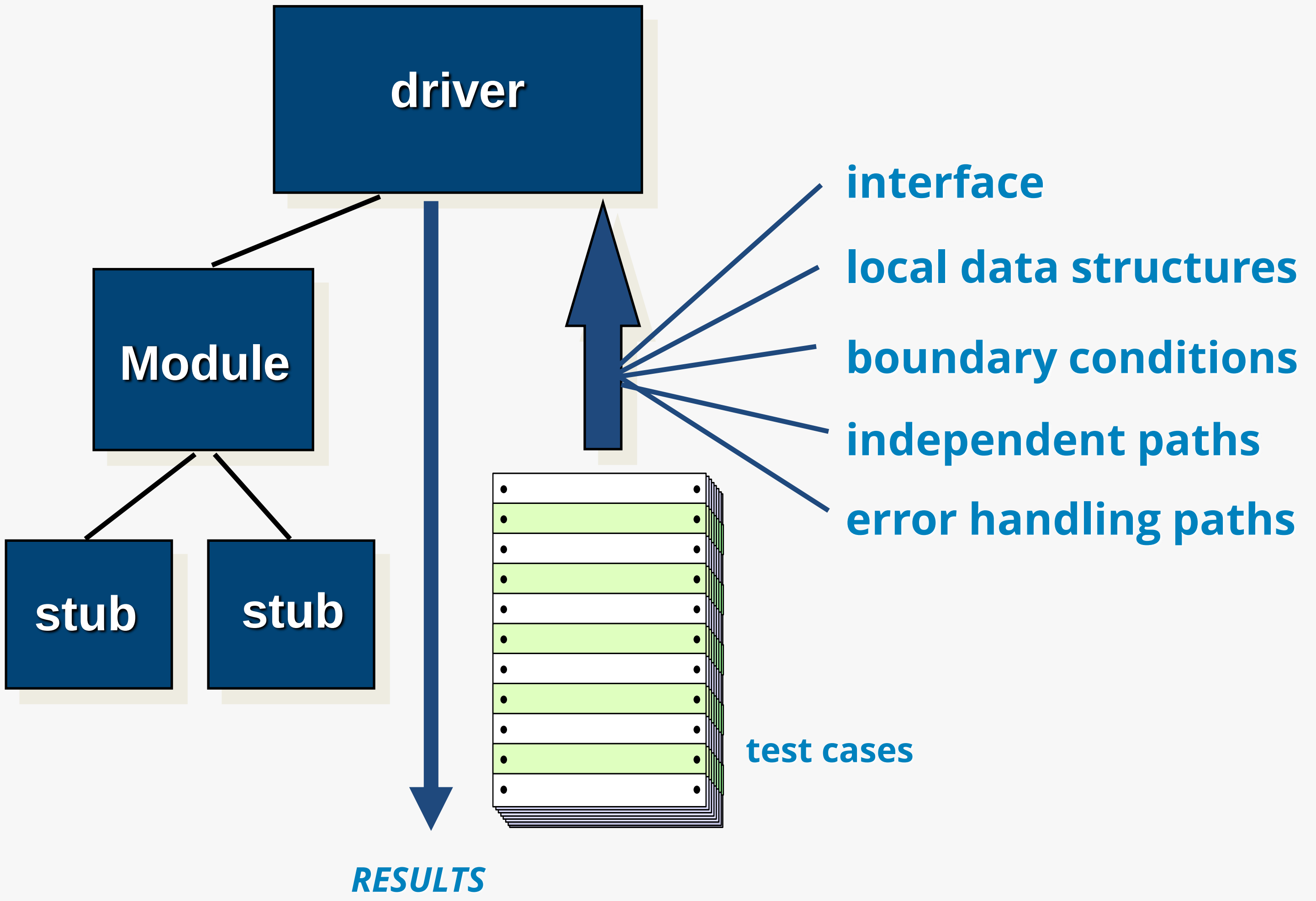
Strategic Testing

Unit Testing



Strategic Testing

Unit Test Environment

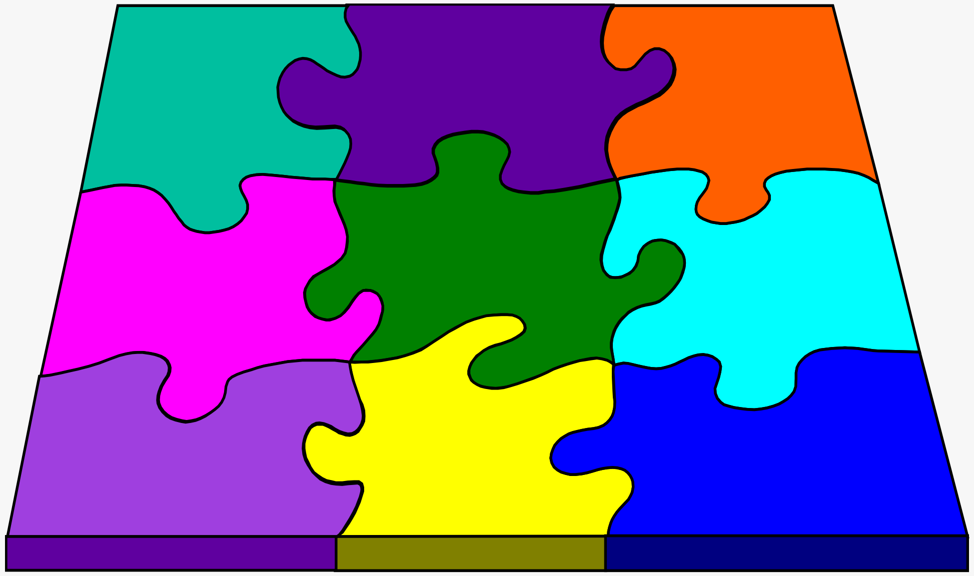


Strategic Testing

Integration Testing Strategies

Options:

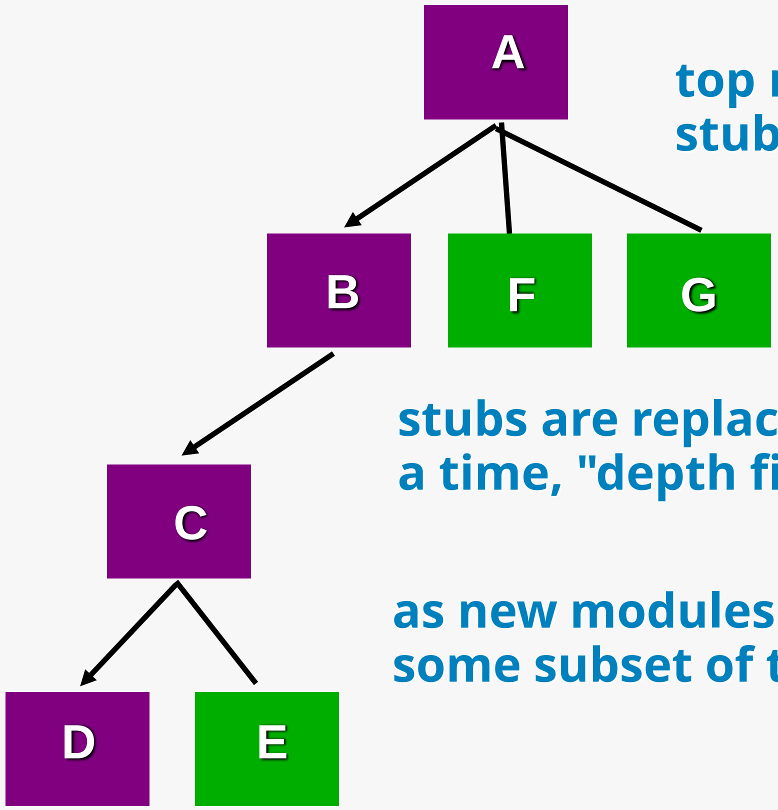
- the “big bang” approach
- an incremental construction strategy



People
Innovation
Excellence

Strategic Testing

Top Down Integration



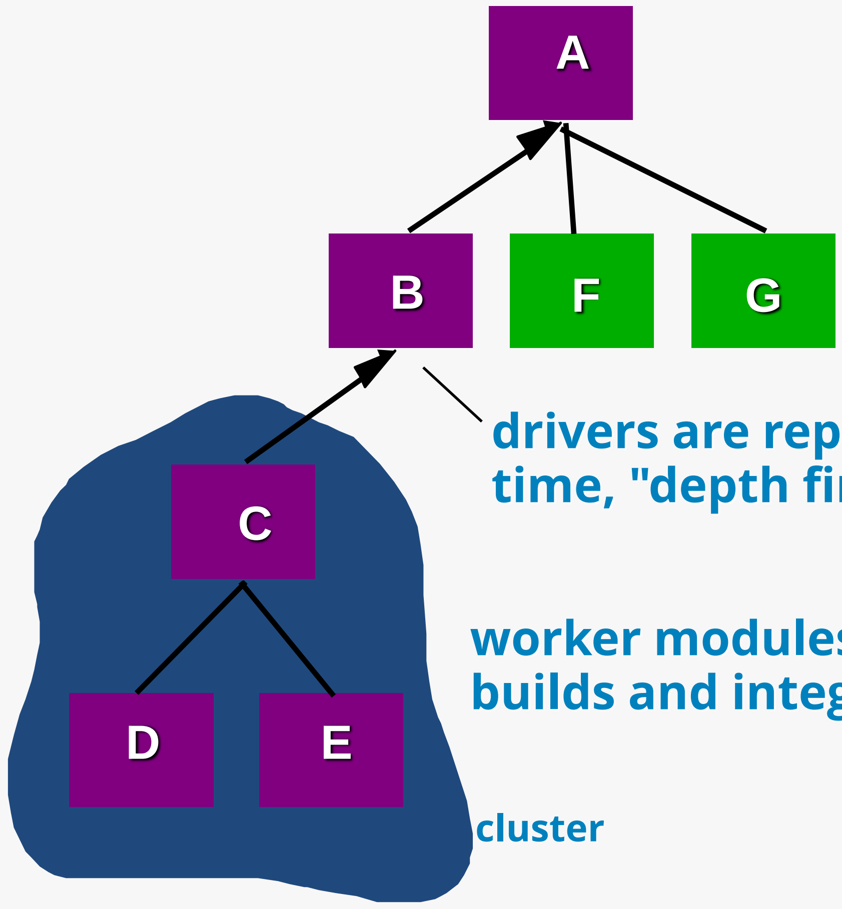
top module is tested with stubs

stubs are replaced one at a time, "depth first"

as new modules are integrated, some subset of tests is re-run

Strategic Testing

Bottom-Up Integration



drivers are replaced one at a time, "depth first"

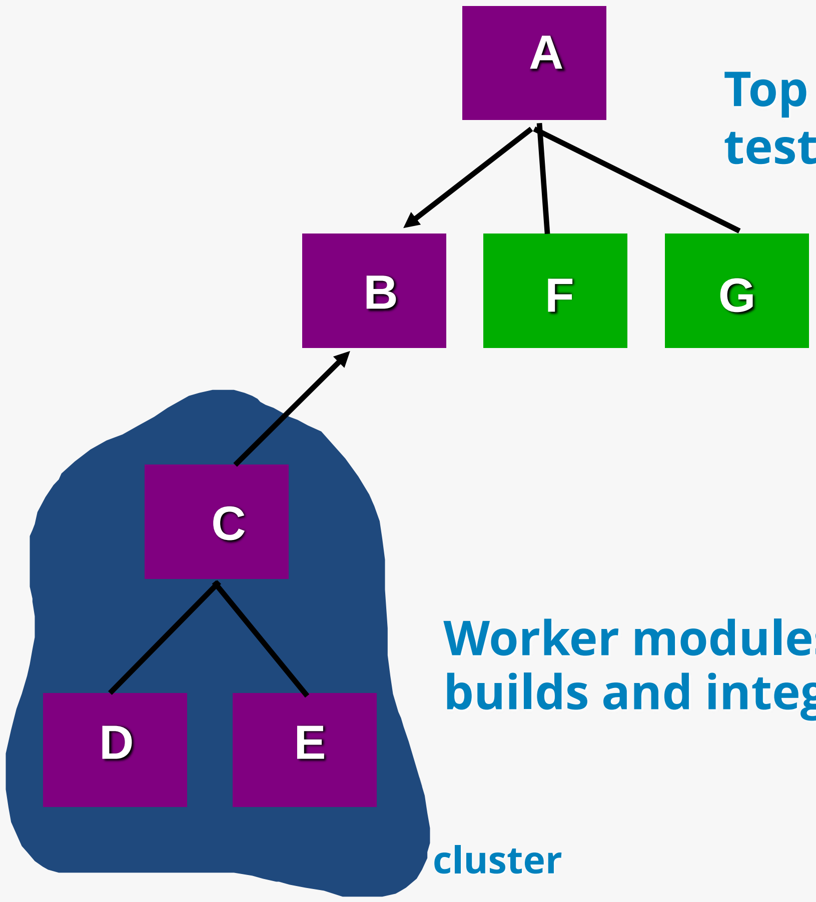
worker modules are grouped into builds and integrated

cluster

People
Innovation
Excellence

Strategic Testing

Sandwich Testing



Top modules are tested with stubs

Worker modules are grouped into builds and integrated

cluster

People
Innovation
Excellence

Strategic Testing

Regression Testing

- *Regression testing* is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects
- Whenever software is corrected, some aspect of the software configuration (the program, its documentation, or the data that support it) is changed.
- Regression testing helps to ensure that changes (due to testing or for other reasons) do not introduce unintended behavior or additional errors.
- Regression testing may be conducted manually, by re-executing a subset of all test cases or using

People
Innovation
Excellence

Strategic Testing

Smoke Testing

- A common approach for creating "daily builds" for product software
- Smoke testing steps:
  - Software components that have been translated into code are integrated into a "build."
    - A build includes all data files, libraries, reusable modules, and engineered components that are required to implement one or more product functions.
  - A series of tests is designed to expose errors that will keep the build from properly performing its function.
    - The intent should be to uncover "show stopper" errors that have the highest likelihood of throwing the software project behind schedule.
  - The build is integrated with other builds and the entire product (in its current form) is smoke tested daily.
    - The integration approach may be top down or bottom

People
Innovation
Excellence

Object-Oriented Testing

- begins by evaluating the correctness and consistency of the analysis and design models
- testing strategy changes
  - the concept of the ‘unit’ broadens due to encapsulation
  - integration focuses on classes and their execution across a ‘thread’ or in the context of a usage scenario
  - validation uses conventional black box methods
- test case design draws on conventional methods, but also encompasses special features

Object-Oriented Testing

Broadening the View of "Testing"

- It can be argued that the review of OO analysis and design models is especially useful because the same semantic constructs (e.g., classes, attributes, operations, messages) appear at the analysis, design, and code level.
- Therefore, a problem in the definition of class attributes that is uncovered during analysis will circumvent side effects that might occur if the problem were not discovered until design or code (or even the next iteration of analysis).

BINUS UNIVERSITY

Object-Oriented Testing

OO Testing Strategy

- class testing is the equivalent of unit testing
  - operations within the class are tested
  - the state behavior of the class is examined
- integration applied three different strategies
  - thread-based testing—integrates the set of classes required to respond to one input or event
  - use-based testing—integrates the set of classes required to respond to one use case
  - cluster testing—integrates the set of classes required to demonstrate one collaboration

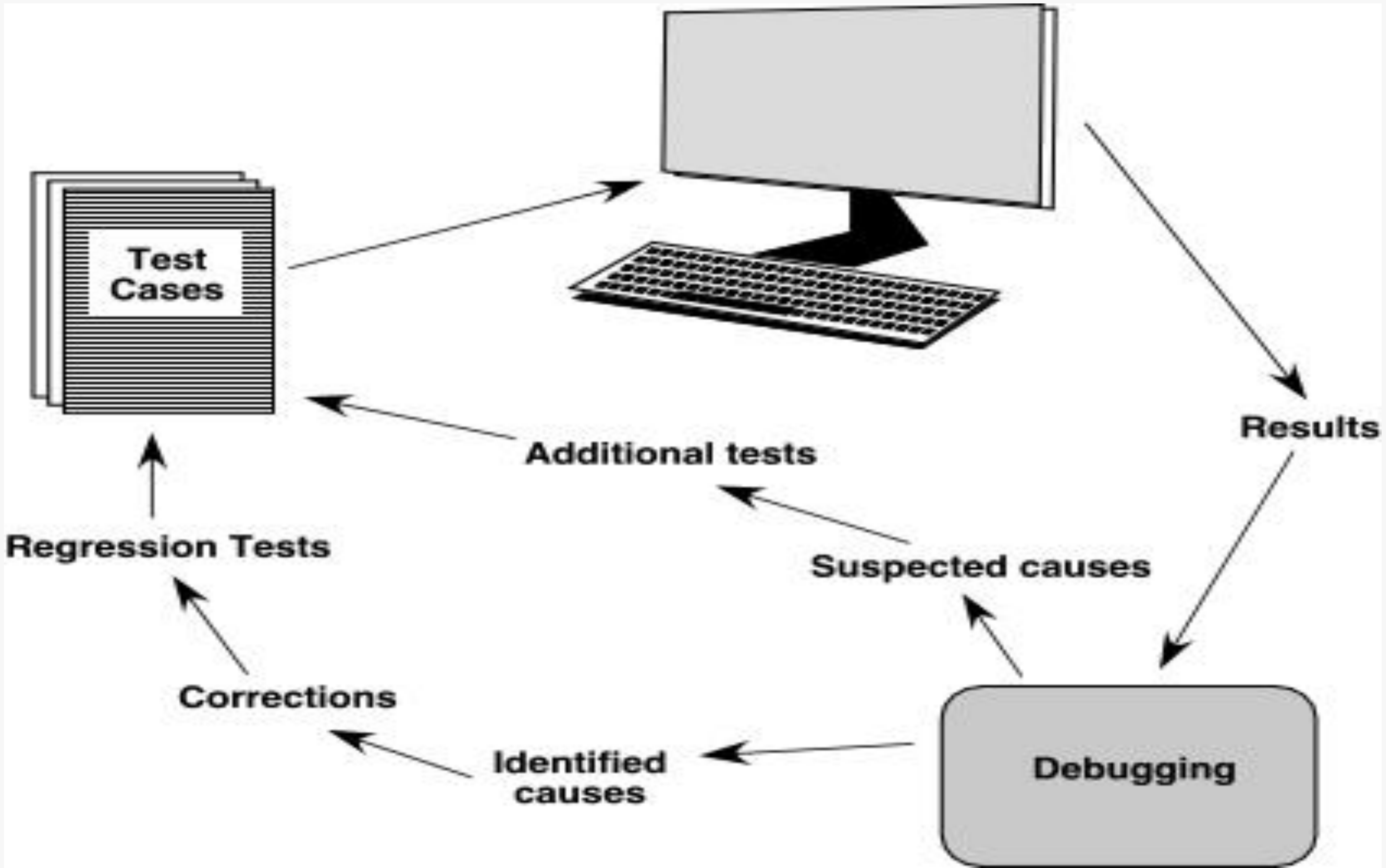
People
Innovation
Excellence

High Order Testing

- Validation testing
  - Focus is on software requirements
- System testing
  - Focus is on system integration
- Alpha/Beta testing
  - Focus is on customer usage
- Recovery testing
  - forces the software to fail in a variety of ways and verifies that recovery is properly performed
- Security testing
  - verifies that protection mechanisms built into a system will, in fact, protect it from improper penetration
- Stress testing
  - executes a system in a manner that demands resources in abnormal quantity, frequency, or volume
- Performance Testing
  - test the run-time performance of software within the context of an integrated system

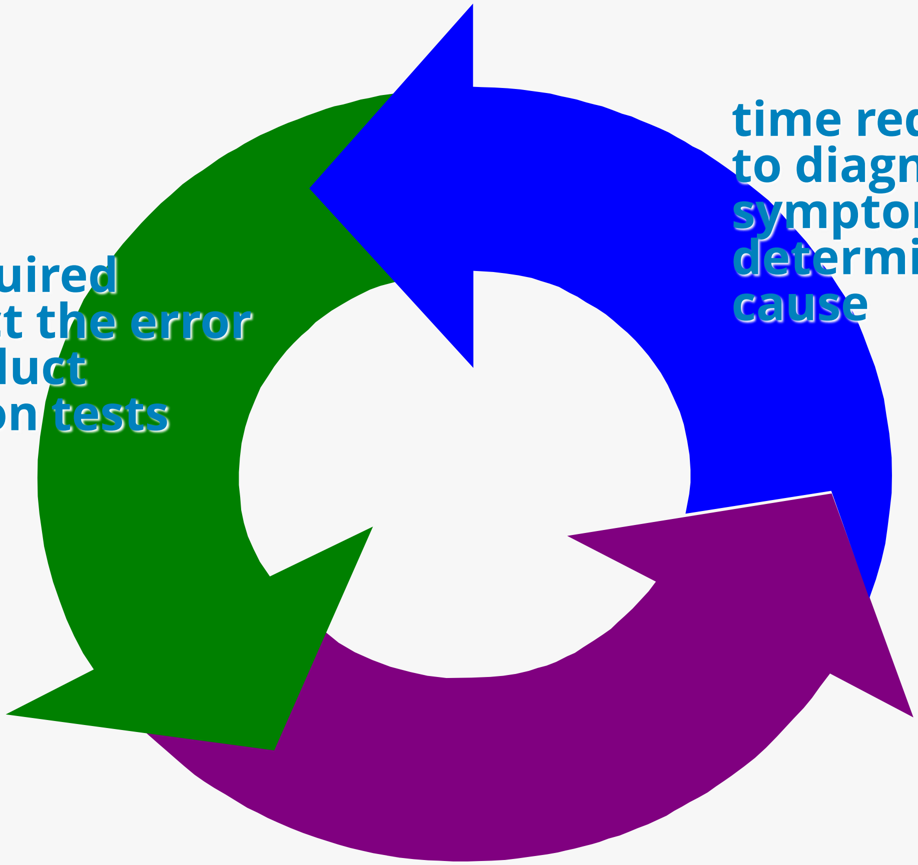
The Art of Debugging

The Debugging Process



The Art of Debugging

Debugging Effort



time required
to correct the error
and conduct
regression tests

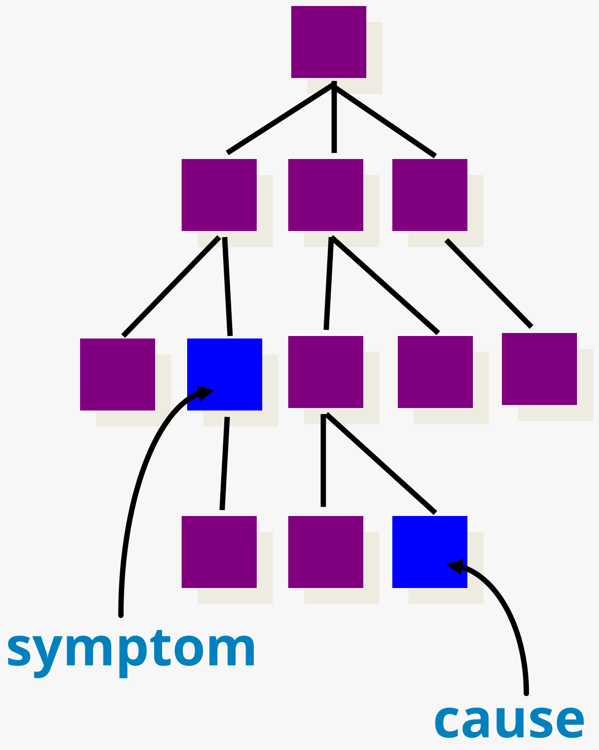
time required
to diagnose the
symptom and
determine the
cause

People
Innovation
Excellence

People
Innovation
Excellence

The Art of Debugging

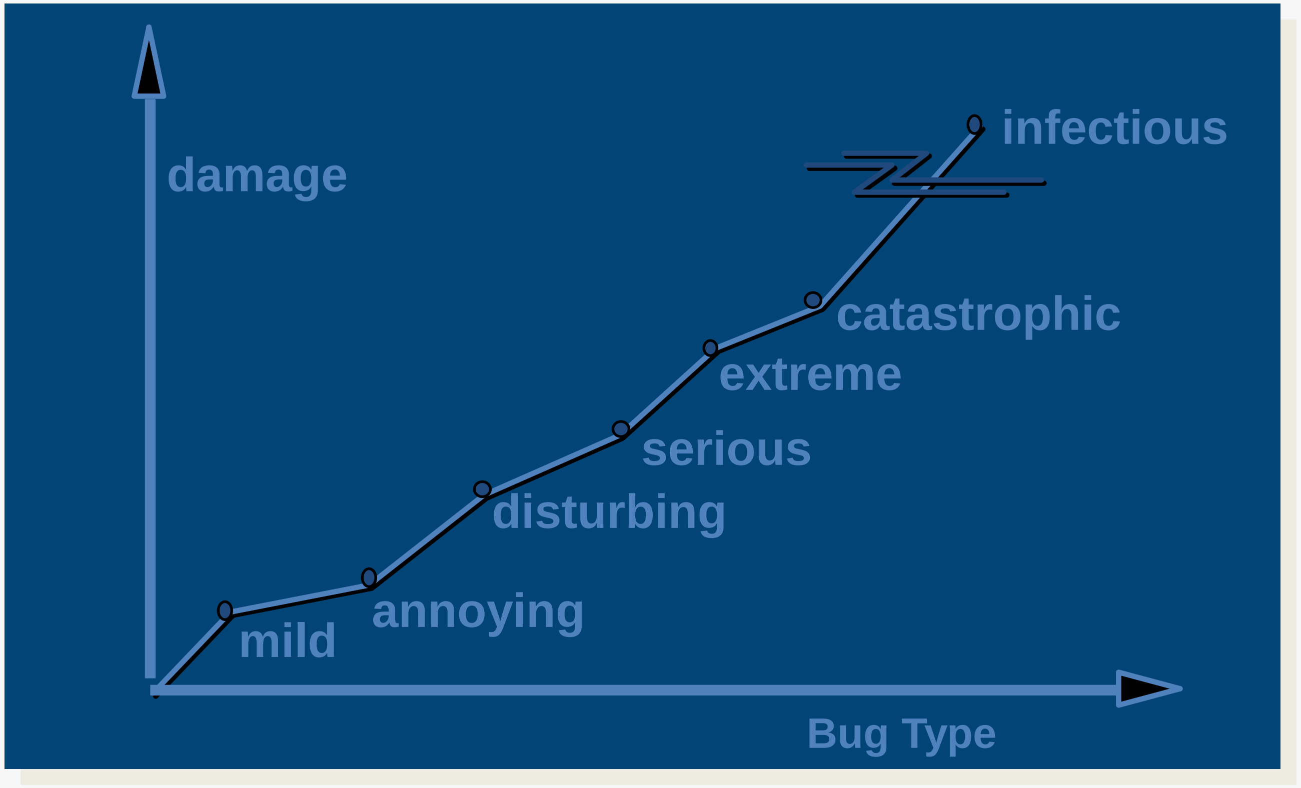
Symptoms & Causes



- symptom and cause may be geographically separated
- symptom may disappear when another problem is fixed
- cause may be due to a combination of non-errors
- cause may be due to a system or compiler error
- cause may be due to assumptions that everyone believes
- symptom may be intermittent

The Art of Debugging

Consequences of Bugs



Bug Categories: **function-related bugs, system-related bugs, data bugs, coding bugs, design bugs, documentation bugs, standards violations, etc.**

The Art of Debugging

Debugging Techniques

- brute force / testing
- backtracking
- induction
- deduction

People
Innovation
Excellence

People
Innovation
Excellence

Exercise

1. Quality and reliability are related concepts but are fundamentally different in a number of ways. Discuss the differences.
2. Besides counting effors and defects, are there other countable characteristics of software that imply quality? What are they and can they be measured directly?
3. What is the difference between alpha and beta testing?
4. How can project scheduling affect integration testing?
5. Who should perform the validation test-the software developer or the software user? Justify your answer

People
Innovation
Excellence

References

- Pressman, R.S. (2015). ***Software Engineering : A Practioner's Approach. 8th ed***. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157.
- **Introduction to software quality assurance, http://www.youtube.com/watch?v=5_cTi5xBlYg**
- **Software reliability , http://www.youtube.com/watch?v=wv51aF_qODA**
- **Software Testing,** http://www.youtube.com/watch?v=caElFKbceP0&list=PLED41984073D5B532

Q & A

People
Innovation
Excellence