Software College Northeastern University
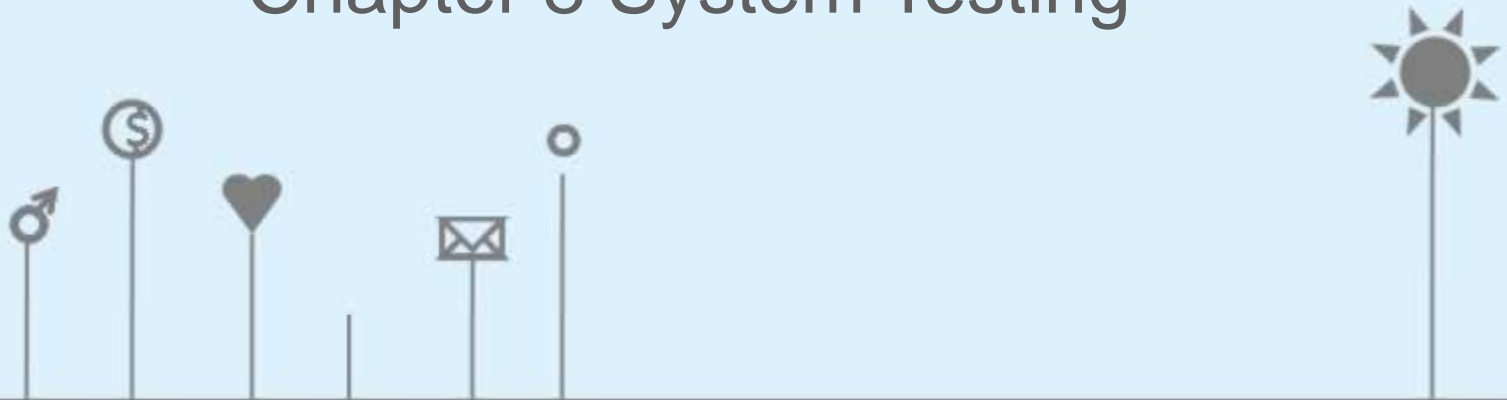
Software Quality Assurance and Testing

Chapter 8 System Testing

wuchenni@qq.com

Contents

Chapter 8

System Testing

8.1 System Testing Concepts

- **System testing**（系统测试） of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

- System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

8.1 System Testing Concepts

- System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.
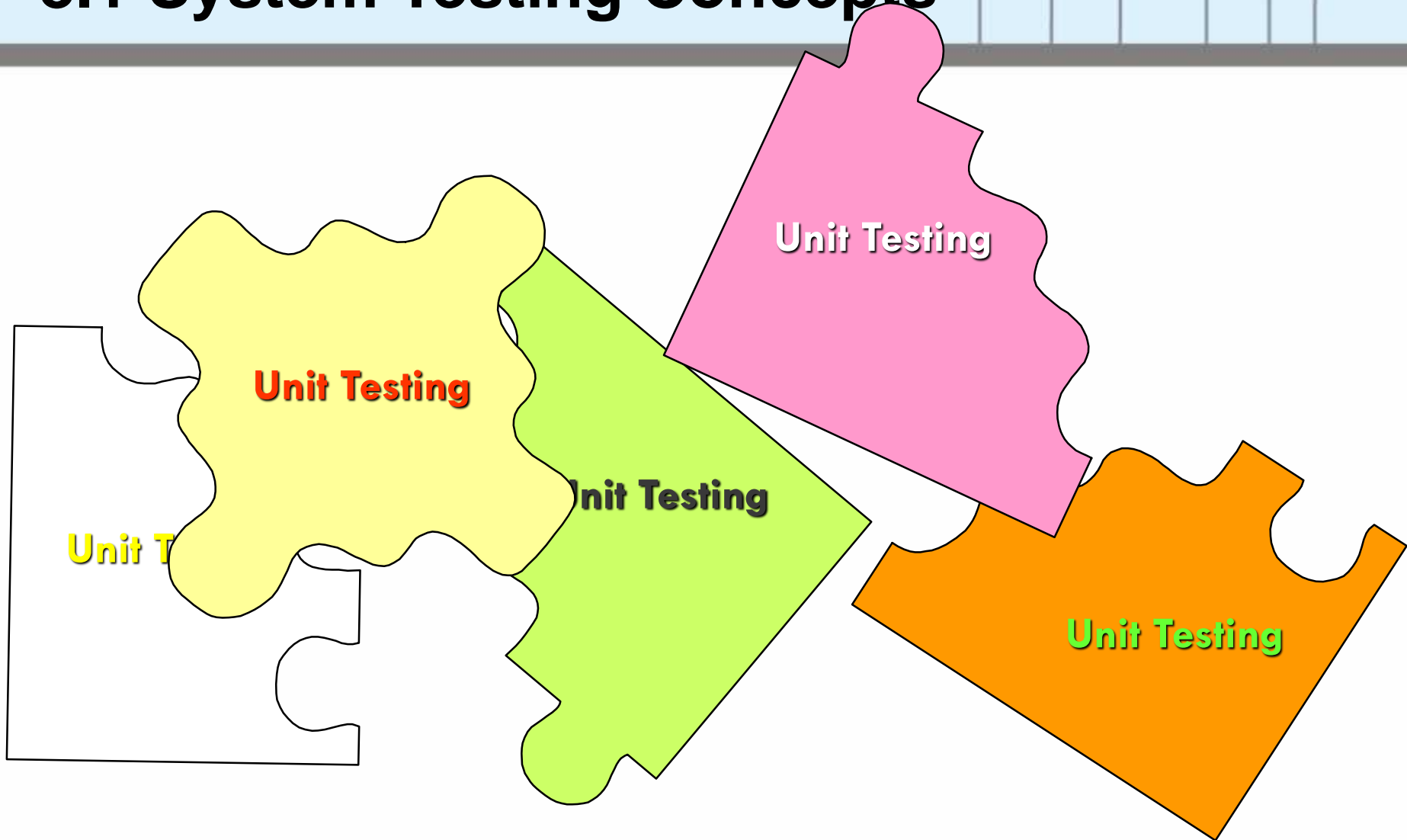
8.1 System Testing Concepts

- System testing is performed on the entire system in the context of a Functional Requirement Specifications (FRS) and/or a System Requirement Specification (SRS).
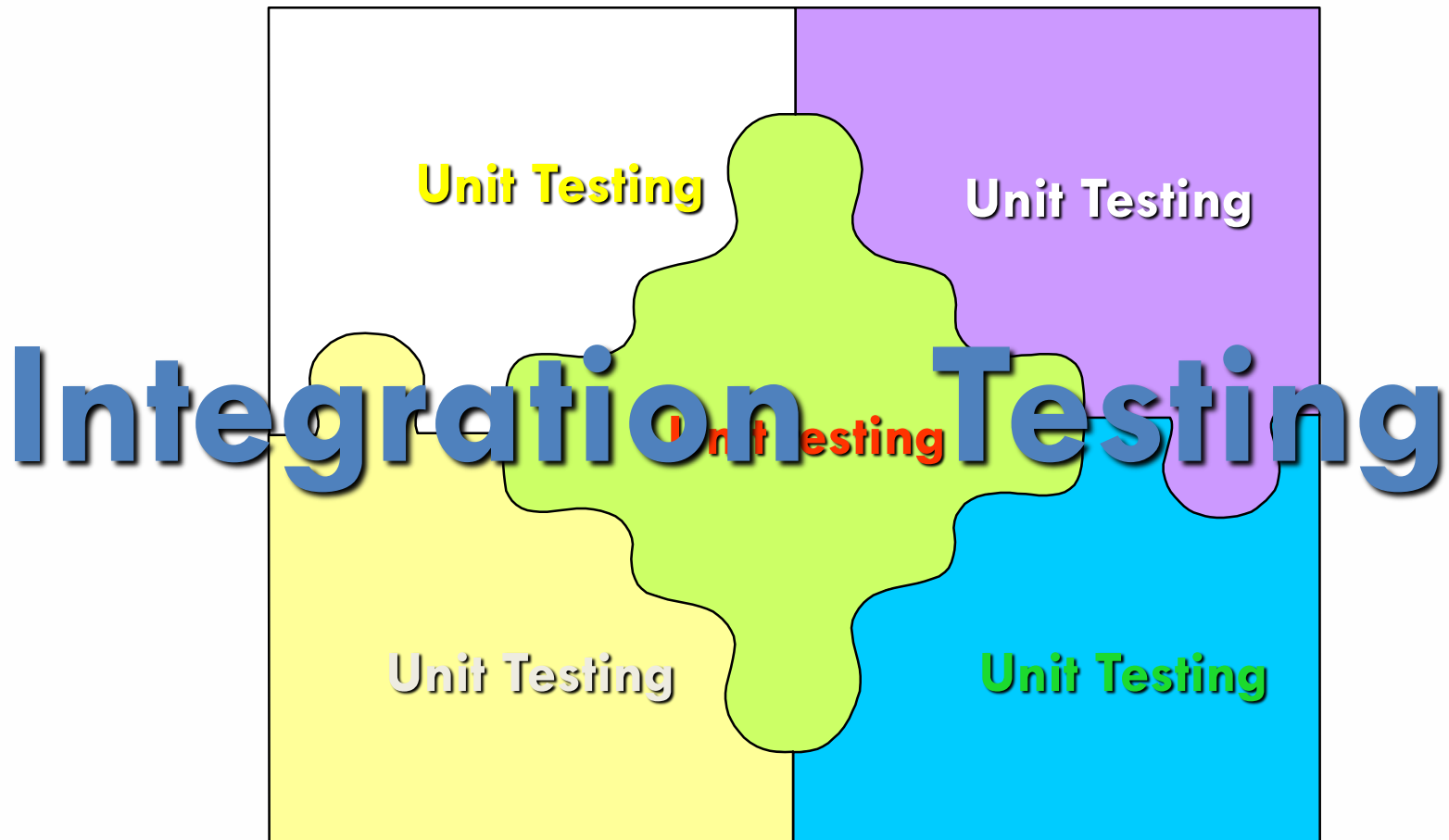
8.1 System Testing Concepts

- **Testing Levels Revisited**
  - Unit testing
    - Verification of isolated software units
  - Integration testing
    - Verification of the interaction among software units
  - System testing
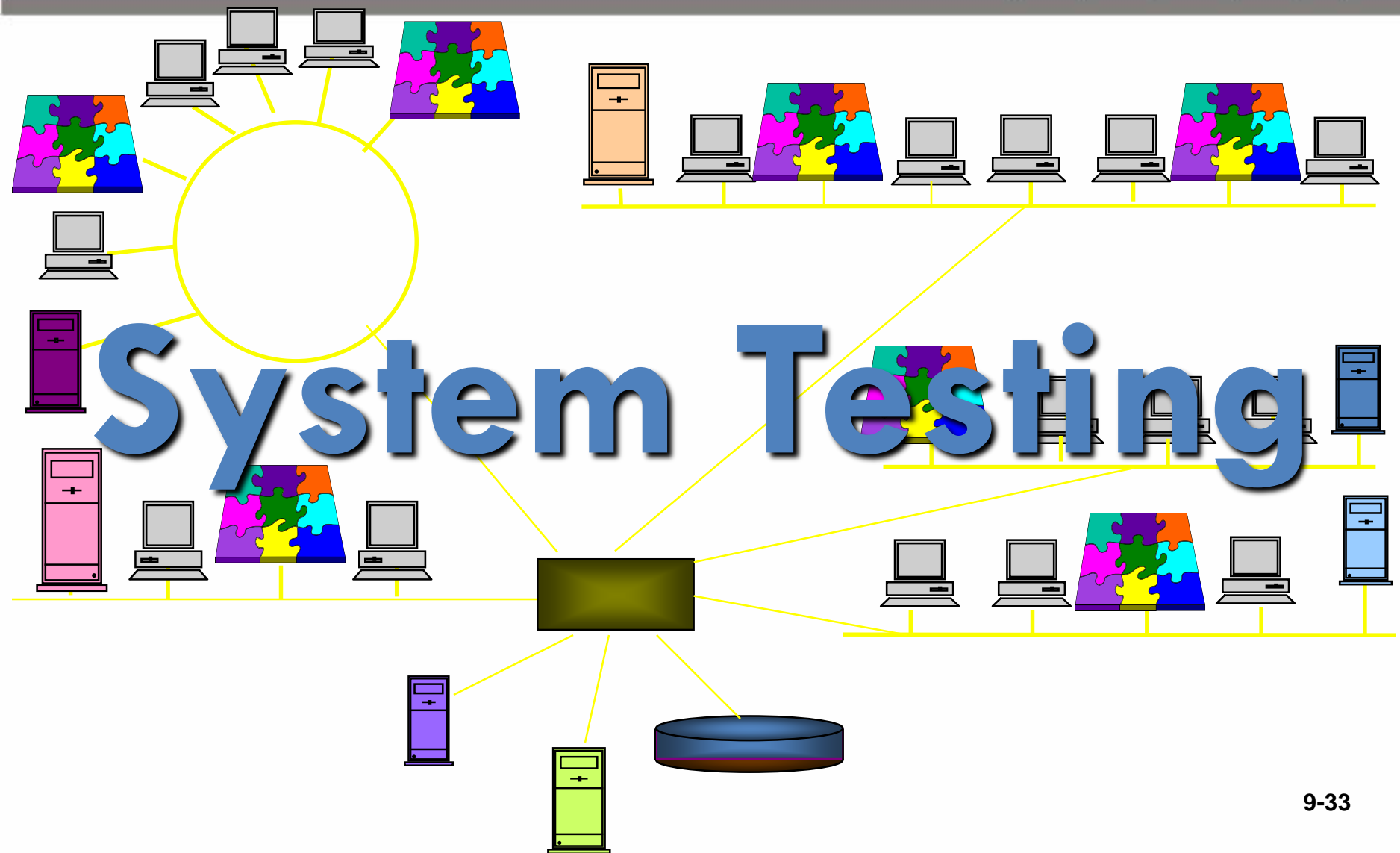   - Verification of the behavior of a whole system

8.1 System Testing Concepts

Unit Testing

Unit Testing

Unit Testing

Unit Testing

Unit Testing

Unit Testing

Unit Testing

Integration Testing
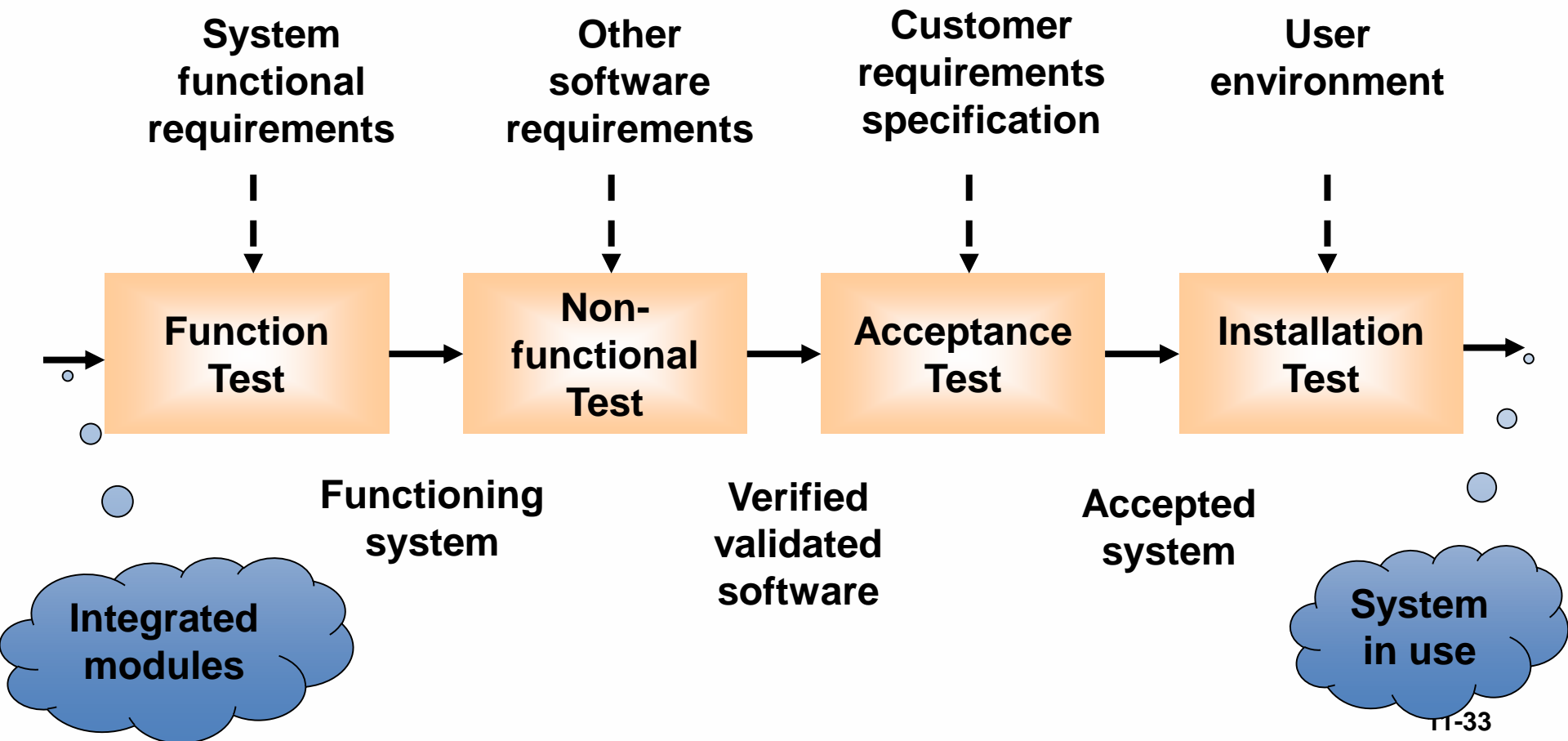
Unit Testing

Unit Testing

Unit Testing

8.1 System Testing Concepts

- **Why system testing is necessary?**
  - Some properties only verifiable at system level
    - Installation , usability, compatibility and maintainability, etc
  - We may involve users at this level
    - Use cases may not map to any specific integration unit
  - The environment of the system is taken into account

8.1 System Testing Concepts

- ## System testing process

System functional requirements	Other software requirements	Customer requirements specification	User environment
↓	↓	↓	↓
Function Test	**Non-functional Test**	**Acceptance Test**	**Installation Test**

Integrated modules → Function Test → **Functioning system** → Non-functional Test → **Verified validated software** → Acceptance Test → **Accepted system** → Installation Test → **System in use**

8.2 System Testing Methods

- 1. GUI software testing
- 2. Usability testing
- 3. Performance testing
- 4. Compatibility testing
- 5. Load testing
- 6. Volume testing
- 7. Stress testing
- 8. Security testing
- 9. Scalability testing
- 10. Sanity testing

- 11. Smoke testing
- 12. Exploratory testing
- 13. Ad hoc testing
- 14. Regression testing
- 15. Reliability testing
- 16. Recovery testing
- 17. Installation testing
- 18. Maintenance testing
- 19. Accessibility testing

- 8.安全性测试：检查系统对非法侵入的防范能力。安全测试期间，测试人员假扮非法入侵者，采用各种办法试图突破防线。

- 9.可扩展性测试：验证代码设计是否良好允许更多的功能在必要时可以被插入到适当的位置中。

- 10.理智测试：软件主要功能成分的简单测试以保证它是否能进行基本的测试。

- 11.冒烟测试：对软件的主要功能进行快餐式测试，最早来自于硬件测试实践，以确定新的硬件在第一次使用时不会着火。用于确认代码中的更改会按预期运行，且不会破坏整个版本的稳定性。

- 12.探索测试：一边测试一边探索，是一个交互式的过程，在某种意义上讲是一个自由形式的测试过程。

8.2 System Testing Methods

- 13.随机测试：测试人员通过随机的尝试系统的功能，试图使系统中断。

- 14.回归测试：发生修改后重新测试先前的测试以保证修改的正确性

- 15.可靠性测试：对系统在规定条件下在指定时间内执行其需要功能的能力进行的测试

- 17.可安装性测试：安装程序是否正确的测试

- 18.可维护性测试：测试系统是否满足可维护性指标，缺陷纠正、性能改进、特性增加引起的

- 19.可访问性测试：可用性（Usability）、可访问性（Accessibility）和可维护（Maintainability）。探求人们怎样才能轻松利用网站以及反馈那些可以改进设计和实施的信息。

8.2 System Testing Methods

- **1.GUI software testing**（图形用户界面测试）
  - GUI testing is the process of testing a product that uses a graphical user interface, to ensure it meets its written specifications.
  - GUI testing includes two parts：one is validation anastomosis(吻合) of interface design and interface implement, the other is to confirm the correctness of the interface processing.

8.2 System Testing Methods

- The difficulties of GUI software testing
  - Possible interface space colossal(巨大的) .
  - Event-driven characteristics
  - GUI test coverage is different from traditional structured coverage, no appropriate automation tools.
  - Bad design makes interface and functional mixed together, the interface of the amendment would lead to more mistakes, but also increase the test difficulty and workload.
  - Interface aesthetics of great subjectivity

8.2 System Testing Methods

- In order to carry on a better GUI test, advocate interface with the functional design of separation.

- Three at designing test cases
  - (1) Division of interface elements and according to the interface of the complexity identify interface hierarchy.
  - (2) In a different interface hierarchy identify the different test strategy.
  - (3) Testing data analysis, extraction test cases.

8.2 System Testing Methods

- **2.Usability testing** （可用性测试）
  - is a technique used to evaluate a product by testing it on users.
  - focuses on measuring a human-made product's capacity to meet its intended purpose.

8.2 System Testing Methods

- **3.Performance testing**
  - Performance testing is used to determine the speed or effectiveness of a computer, network, software program or device.
  - can involve quantitative tests done in a lab
  - Qualitative attributes such as reliability, scalability and interoperability may also be evaluated.
  - Performance testing is often done in conjunction with stress testing.
  - The goals of performance testing are to eliminate bottlenecks and improve system reliability.

8.2 System Testing Methods

- ## 7.Stress testing

  - refers to tests that put a greater emphasis on robustness, availability, and error handling under a heavy load, rather than on what would be considered correct behavior under normal circumstances.

  - is designed to stress the system to failure.

  - The goal of stress testing is increased recoverability.

8.2 System Testing Methods

- This type of testing in the way need abnormal quantity, frequency or resources under execution system.

- Stress testing belongs to boundary testing.

- One of it is called the sensitivity test technology.

- Research system in short time in response to the peak activities.

8.2 System Testing Methods

- **6.Volume testing**
  - Volume testing refers to testing a software application with a certain amount of data.
  - Volume testing is facing to data, its purpose is to show the system can handle the data amount confirm the target.
  - This amount can
    - be the database size
    - be the size of an interface file

8.2 System Testing Methods

- Methods and steps：
  - (1) Analysis the system of external data and classification,
  - (2) To the data source for each type analysis the limitation of volume,
  - (3) For the data source for each type prepare large volume of data and test the system,
  - (4) Analyze test result, and compared with expectations determine currently system volume bottlenecks,
  - (5) Optimization the system, and repeat the steps until the system achieve expected volume processing ability.

8.2 System Testing Methods

- **5.Load testing**
  - is the process of putting demand on a system or device and measuring its response.
  - is most relevant for multi-user systems, often one built using a client/server model, such as web servers.
  - When the load placed on the system is raised beyond normal usage patterns, in order to test the system's response at unusually high or peak loads, it is known as stress testing.

8.2 System Testing Methods

- **4.Compatibility testing**（兼容性测试）
  - Compatibility testing is part of software non-functional tests, conducted on the application to evaluate the application's compatibility with the computing environment.

8.2 System Testing Methods

- Computing environment may contain some or all of the below mentioned elements:
  - Computing capacity of hardware platform (IBM 360, HP 9000, etc.)..
  - Bandwidth handling capacity of networking hardware
  - Compatibility of peripherals (Printer, DVD drive, etc.)
  - Operating systems (MVS, UNIX, Windows, etc.)
  - Database (Oracle, Sybase, DB2, etc.)
  - Other system software (Web server, networking/ messaging tool, etc.)
  - Browser compatibility (Firefox, Netscape, Internet Explorer, Safari, etc.)
  - Carrier compatibility (Verizon, Sprint, Orange, O2, AirTel, etc.)
  - Backwards compatibility. （向后兼容）
  - Different compilers (compile the code correctly)
  - Runs on multiple host/guest Emulators

8.2 System Testing Methods

- **16.Recovery testing**
  - Recovery testing is the activity of testing how well the application is able to recover from crashes, hardware failures and other similar problems.

8.2 System Testing Methods

- In some cases, the system must be tolerant, running process error cannot make the whole system function have stopped.

- In other cases, a system of mistake must be in a certain period to correct, or can produce severe economic losses.

- Recovery testing use all sorts of manual intervention means making software error, validation system recovery ability.

- If the system itself can auto reply, inspection recovery and restart is correct.

- If this recovery will take manual intervention, it should consider whether the mean time to repair within the limited scope.

8.2 System Testing Methods

- Examples of recovery testing:
  - While the application running, suddenly restart the computer and after that check the validness of application's data integrity;
  - While application receives data from the network, unplug and then in some time plug-in the cable, and analyze the application ability to continue receiving of data from that point, when network connection disappeared;
  - To restart the system while the browser will have definite number of sessions and after rebooting check, that it is able to recover all of them.
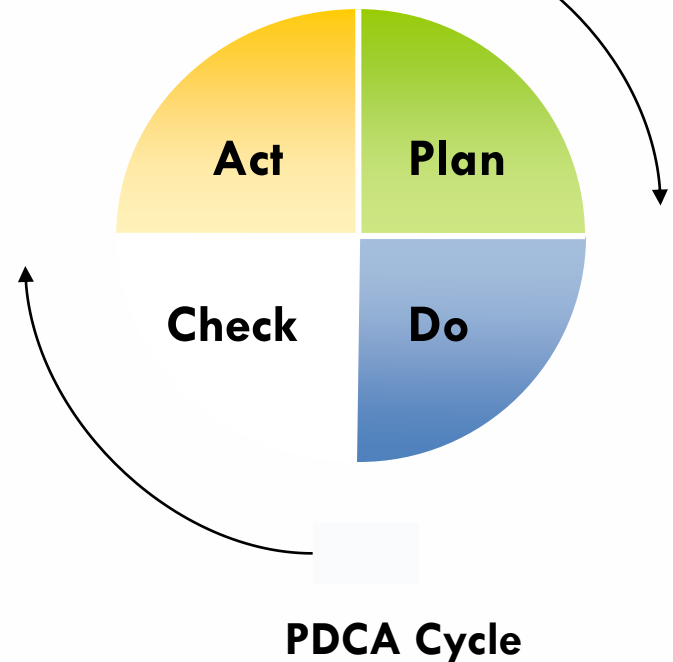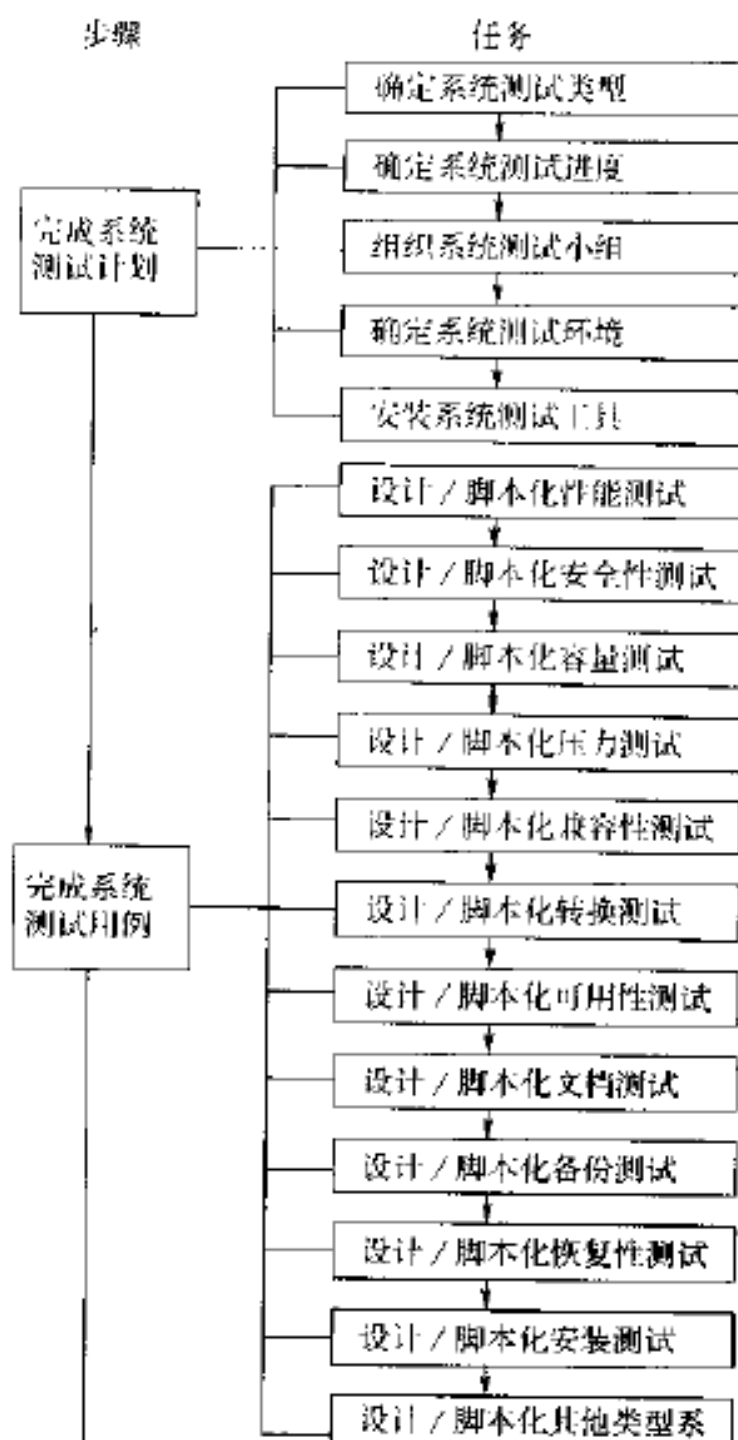
8.3 System Testing process

- System testing evaluate the whole application functionality and performance.

- Similar to the unit testing and the integrated testing, system test also need to follow certain process.

- Deming cycle
  - Shewhart puts forward
  - Deming introduction to Japan
  - Applied to the software quality — PDCA process

8.3 System Testing process

- Plan：
  - People define goals, and sure to get these goals and need conditions and methods.

- Do：
  - Do work according to plan

- Check：
  - Through the examination to determine whether it is scheduled

- Act：
  - If no scheduled, or does not conform to the expected, choose suitable action for correction

Act   Plan

Check   Do

PDCA Cycle

8.3 Syste

步骤 | 任务

完成系统
测试计划
- 确定系统测试类型
- 确定系统测试进度
- 组织系统测试小组
- 确定系统测试环境
- 安装系统测试工具

完成系统
测试用例
- 设计／脚本化性能测试
- 设计／脚本化安全性测试
- 设计／脚本化容量测试
- 设计／脚本化压力测试
- 设计／脚本化兼容性测试
- 设计／脚本化转换测试
- 设计／脚本化可用性测试
- 设计／脚本化文档测试
- 设计／脚本化备份测试
- 设计／脚本化恢复性测试
- 设计／脚本化安装测试
- 设计／脚本化其他类型系