

第3章 功能测试自动化

■ 教学目标

- 理解功能测试自动化要点
- 初步掌握自动化测试工具的选型

■ 重点

- 自动化测试原理
- 自动化测试用例设计与脚本开发
- 自动化测试的流程



- 3.1 功能测试自动化要点
- 3.2 软件自动化测试工具



小组讨论，选出一位代表发言

你想象中的功能自动化测试是什么样？

你认为功能自动化为什么会被提出？

你认为功能自动化测试的流程和手工测试流程区别点在那里？

你认为什么样的场景适合开展自动化测试？（何时、何地、何人）

在脑海中想象一下，如果让你对一个项目进行自动化测试，你会遇到哪些难点或者问题？



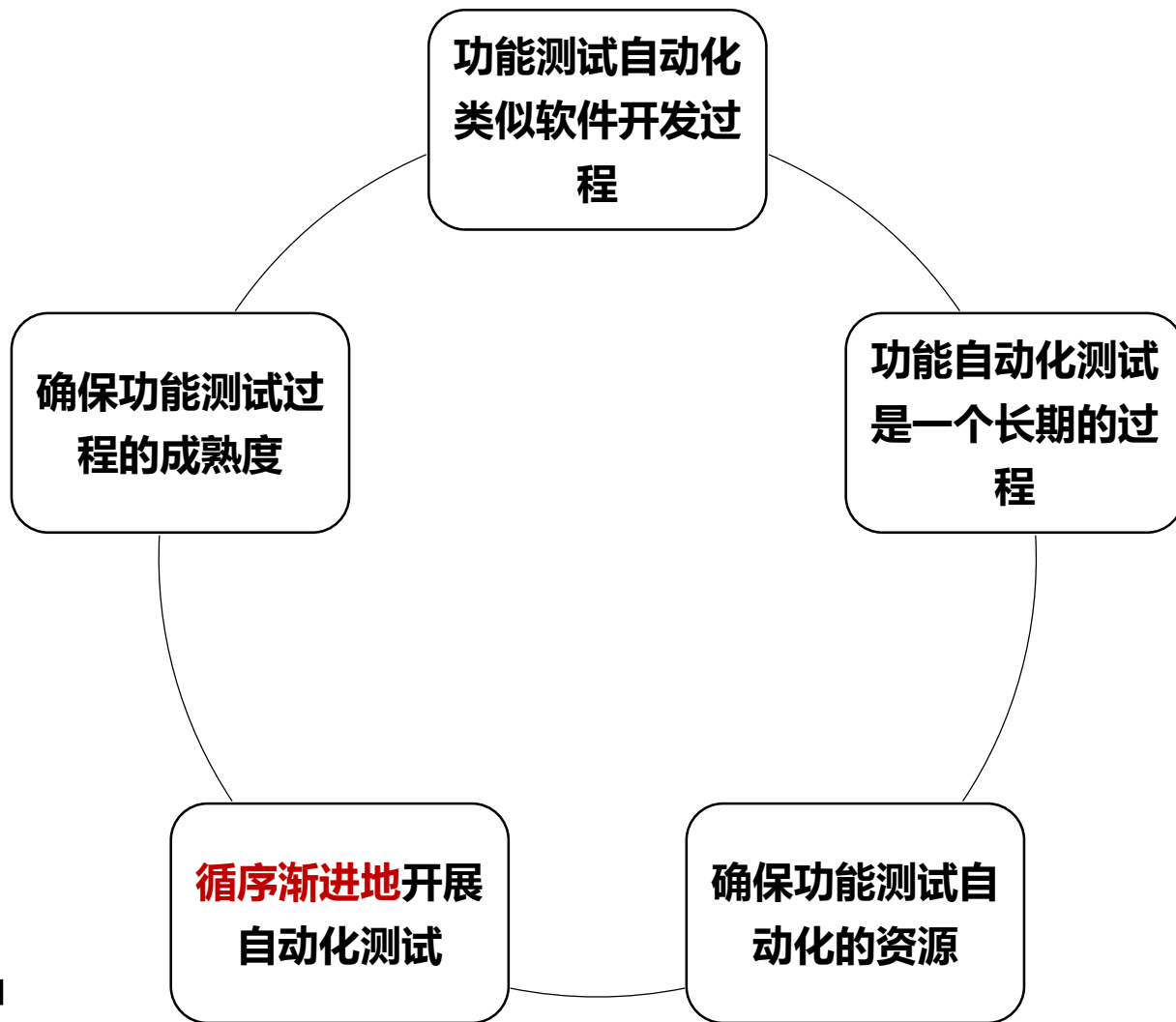
3.1 功能测试自动化要点

- **手工测试 VS. 自动测试**
 - 重复测试
 - 准确性问题
 - 高效性
- **手工测试具有自动化测试工具不可替代的作用**
 - 测试用例的设计
 - 界面和用户体验测试
 - 正确性的检查



3.1.1 何时开始使用自动化测试

- 成功开展自动化测试考虑的5个方面



3.1.2 如何开展自动化测试



选取合适的测试项目来开展自动化测试

选择合适的自动化测试开始的时间

选择合适的自动化测试工程师，构建测试团队

控制自动化测试的成本

选择合适的时机，展开自动化相关工作，但准备工作要始于项目前期

3.1.2 自动化测试的一些误区

■ 常见误区



期望自动化测试完全替代手工测试，能实现100%的测试覆盖率

期望自动化测试发现大量新的缺陷

不关注前期测试设计，而单纯指望引入工具就能做好自动测试

自动测试是后期的工作，前期单元、集成测试做不做，做得深不深无所谓

自动化测试工具是“万能”的，可用于所有测试，并使测试工作量迅速减少

自动化测试顺利引入后就一劳永逸了

3.1.2 适合自动化测试的场景

■ 适用场景



3.1.2 自动化测试的优点

■ 优点



可以对程序的新版本自动执行回归测试

可以执行一些手工测试困难或不可能进行的测试

可以更好地利用资源

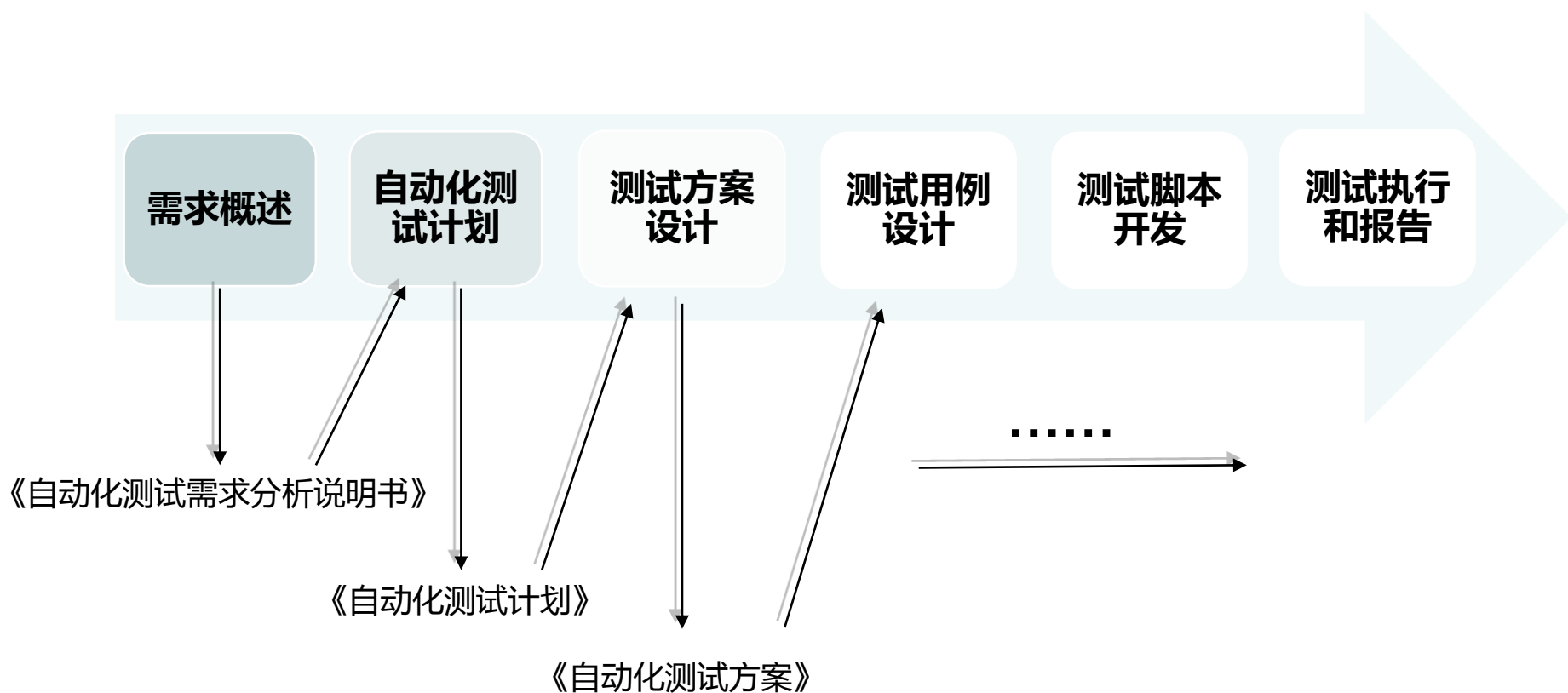
测试具有一致性和可重复性

可以更快地将软件推向市场

可以增加软件的信任度

3.1.3 自动化测试项目的流程

■ 自动化测试项目的主要流程



3.1.3 自动化测试项目的流程

需求分析

测试计划

测试点/测试例

手工测试

缺陷报告

测试报告

需求分析

测试计划

测试点/测试例

编写脚本

执行脚本

缺陷报告

测试报告

■ 自动化测试方案考虑的因素

- 项目的影响
- 复杂度
- 时间
- 早期需求和代码的稳定性
- 维护工作量
- 覆盖率
- 资源
- 自动化测试的执行

1. 自动化测试与手工测试相比的优势有哪些？

答案：

- 重复性
- 准确性
- 高效性

2. 用自己的理解解释自动化测试项目的流程。

答案：

- 需求概述
- 自动化测试计划
- 测试方案设计
- 测试用例设计
- 测试脚本开发
- 测试执行和报告



■ 自动化测试用例三部曲

➤ 筛选手工测试用例的过程

- 筛选出所有要被“测试自动化”的手工测试用例，筛选出来可以直接使用的用例和需要转换之后才能使用的用例

➤ 转换手工测试用例的过程

- 按照模板修改测试用例，使之符合自动化测试用例的格式和规则
- 优化自动化测试业务流程（更精简、严格）

➤ 新增和补充自动化测试用例的过程

- 主要是新增和补充一些手工测试用例没有涉及的自动化测试业务流程
- 相比于手工用例，建议自动化测试用例步骤写的**详细一些**，更加友好地指导脚本开发

■ 自动化测试用例设计原则

- 自动化测试用例的范围往往是核心业务流程或者重复执行率较高的
- 自动化测试用例的选择一般以“正向”为主
 - 删去不重要的反向用例，减少脚本繁琐程度
 - 功能自动化测试主要用于回归测试，回归测试的目的就是保证新增功能后老功能是否能正常继续运作
- 不是所有的手工测试用例都可以使用自动化测试来实现
 - 如涉及到一些硬件辅助的操作
- 手工测试用例可以不回归原点，而自动化用例往往是必须的
- 自动化测试用例和手工测试用例不同，不需要每个步骤都写预期结果

■ 自动化测试脚本分类

- 线性脚本：通过录制直接产生的线性执行的脚本
- 结构化脚本：具有顺序、循环、分支等结构的脚本
- 可共享脚本：可以被多个测试用例使用，被其他脚本调用的脚本
- 数据驱动脚本：测试数据和业务流程控制分离的脚本，通过读入数据文件来驱动流程进行的脚本
- 关键字驱动脚本：脚本、数据、业务分离，看起来更像是描述一个测试用例在做什么，而不是如何做

■ 自动化测试脚本开发注意

- 自动化测试脚本代码必须严谨、规范
- 自动化测试脚本是参照自动化测试用例开发出来的，测试用例即是开发参照物
- 发挥自己的想象尽一切可能使自动化测试脚本更智能、高效、稳定、复用性高
- 开发过程中多用技术手段如断点、调试等检查业务组件是否存在缺陷或代码是否存在漏洞
- 自动化测试脚本开发完毕后，至少运行成功3次以上，方可认为脚本已经没有问题

■ 后期我们关注的几个具体技术点：

➤ 计算机是怎么识别对象的？

- 对象属性、对象库

➤ 脚本是怎么进行检查对错的？

- 检查点

➤ 脚本怎么能够进行复用、模块化？

- 参数化、函数库



■ 自动化功能测试工具

- 实现了自动化执行测试用例，自动化地检查测试数据的测试工具，替代人工进行测试步骤的执行，从而验证应用程序是否满足了特定功能的测试工具。

3.2.1 自动化测试工具选型

测试工具评估

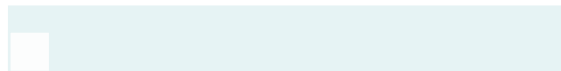


- ☐ 分析项目特点
- ☐ 目前存在的各种工具的情况
- ☐ 编写选型报告

测试工具试用



自动化测试工具培训



- ☐ 测试工具的总体介绍
- ☐ 测试工具操作方法
- ☐ 测试工具使用实践
- ☐ 对测试工具相关测试理论进行讲解



■ 1. 自动化工具的选型要考虑哪些因素？

答案：

1) 测试工具评估

- 分析项目的特点
- 同时还要了解目前存在的各种测试工具的情况
- 选型的最后一步是编写选型报告

2) 测试工具试用

3) 自动化测试工具培训

- 测试工具的总体介绍。
- 测试工具操作方法。
- 测试工具使用实践。
- 对测试工具相关的测试理论进行讲解。

■ 2. 用自己的话谈谈你对基于GUI对象的自动化测试工具的理解。

■ 3. 自动化测试用例设计原则？

- 自动化测试用例的范围往往是核心业务流程或者重复执行率较高的
- 自动化测试用例的选择一般以“正向”为主
- 不是所有的手工测试用例都可以使用自动化测试来实现
- 手工测试用例可以不回归原点，而自动化用例往往是必须的
- 自动化测试用例和手工测试用例不同，不需要每个步骤都写预期结果

■ 什么是框架framework ?

- ❖ 整个或部分系统的**可重用**设计，表现为一组抽象构件以及构件实例间交互的方法；
- ❖ 可被开发者定制的应用骨架。

前者是从**应用**方面、而后者是从**目的**方面给出的定义。测试框架也是如此，测试框架出现的最终目的是花少量的资源来完成尽可能多的测试任务，所以测试框架的建立以及框架的重用性方面是最值得测试人员深入探究的地方。

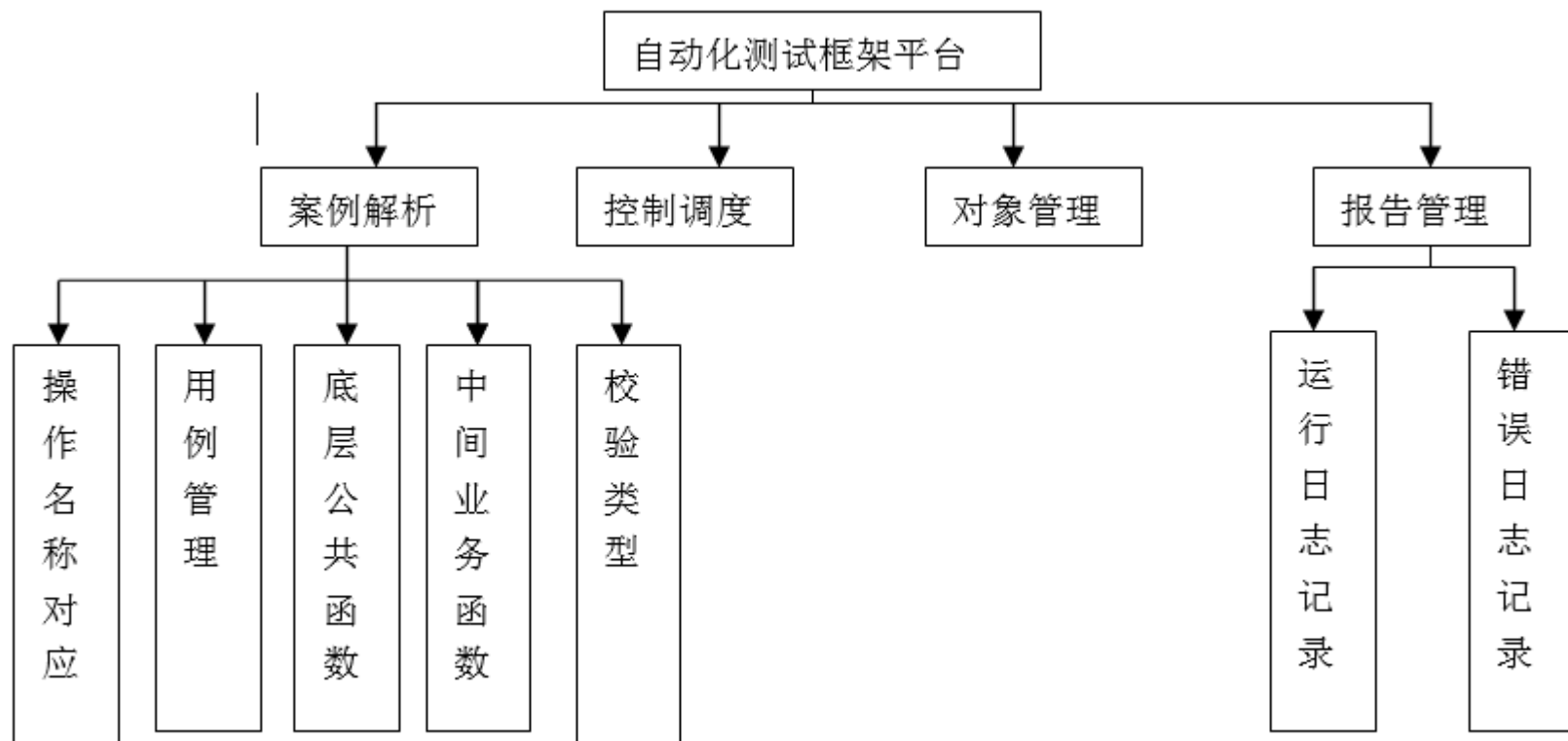
■ 什么是测试框架？

测试框架是一组自动化测试的规范、测试脚本的基础代码，以及测试思想、惯例的集合。

■ 测试框架的好处在于：

- ❖ 减少冗余代码、提高代码生产率、提高代码重用性和可维护性。提高开发速度，提升测试代码的执行效率；
- ❖ 提高软件代码质量，同时引入重构概念，让代码更干净和富有弹性；
- ❖ 提升系统的可信赖度，作为回归测试的一种实现方法支持修复后“再测试”，确保代码的正确性。

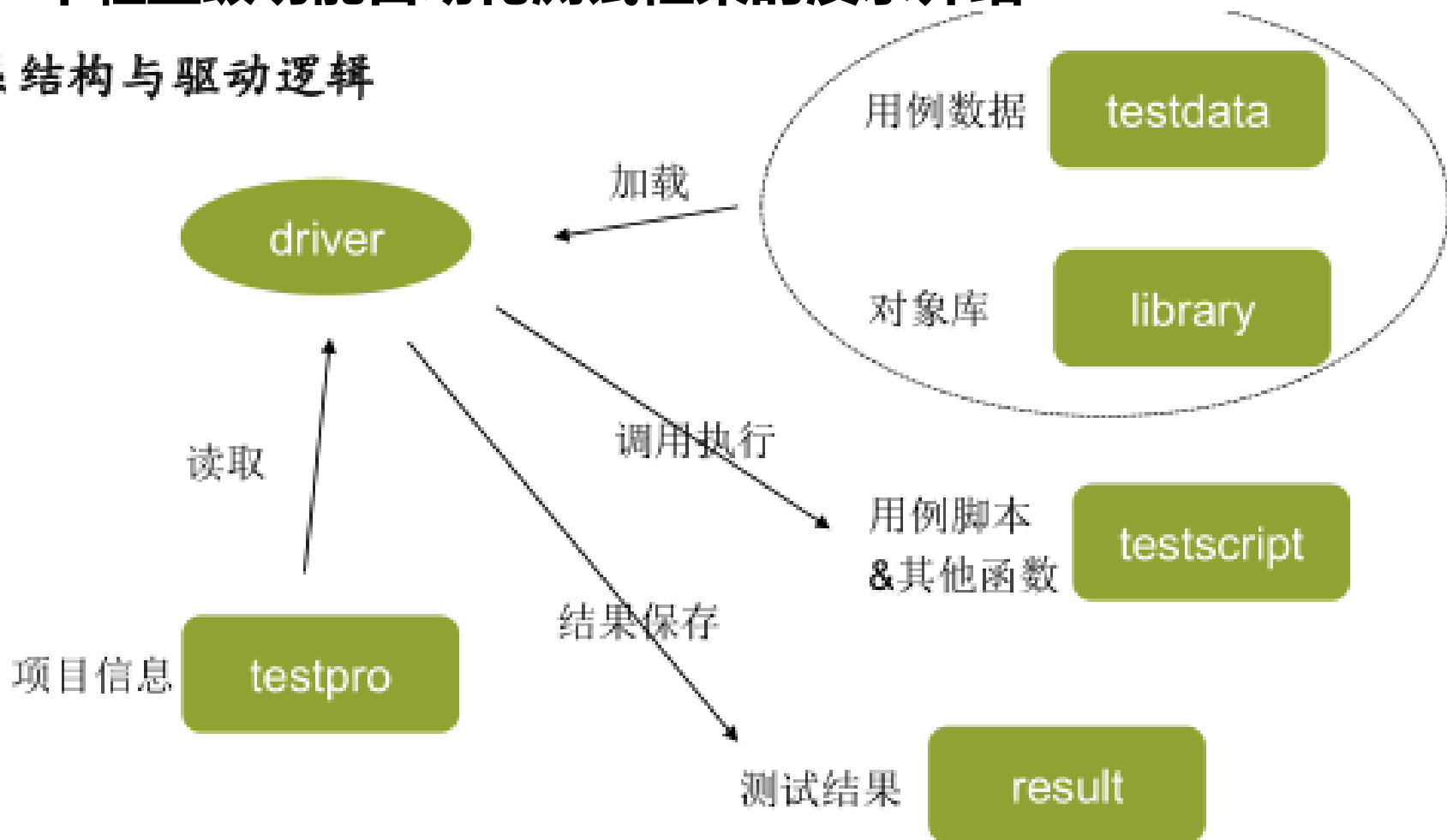
■ 功能自动化测试框架全景图



3.3 自动化测试框架

■ 一个轻量级功能自动化测试框架的演示介绍

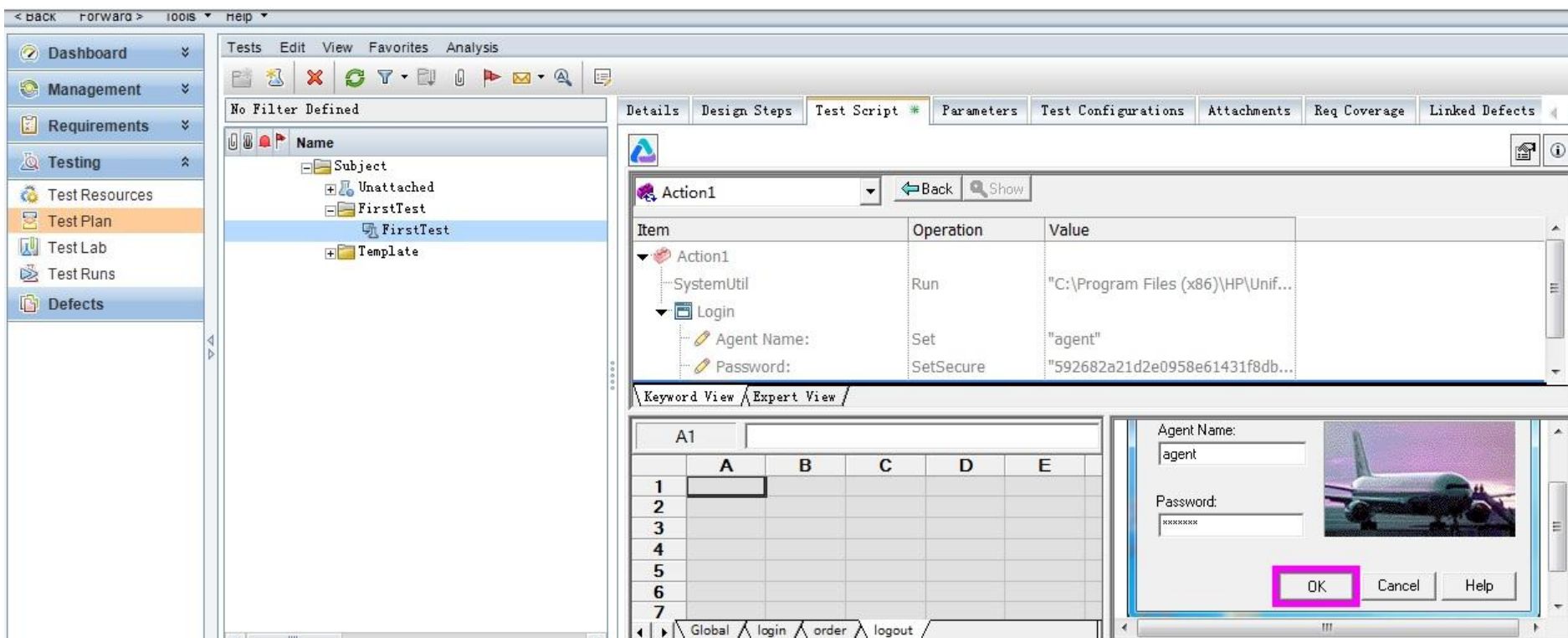
体系结构与驱动逻辑



3.3 自动化测试框架

■ UFT & ALM 集成演示

- 可以在ALM中直接创建测试，也可以在UFT创建测试上传至ALM
- ALM可以管理测试中的各种资源，可以远程执行测试
- ALM 与 UFT 结合，可以实现HP BPT测试框架，实现完全的业务、数据、脚本的分离，BPT内容在第11章有更多介绍



The screenshot displays the HP UFT/ALM interface. On the left is a navigation pane with categories like Dashboard, Management, Requirements, Testing, Test Resources, Test Plan, Test Lab, Test Runs, and Defects. The 'Testing' section is expanded, showing a tree structure with 'Subject', 'Unattached', 'FirstTest', and 'Template'. The main area shows the 'Test Script' tab for 'Action1'. It contains a table with columns 'Item', 'Operation', and 'Value'.

Item	Operation	Value
SystemUtil	Run	"C:\Program Files (x86)\HP\Unif..."
Login		
Agent Name:	Set	"agent"
Password:	SetSecure	"592682a21d2e0958e61431f8db..."

Below the table is a 'Keyword View' section with a grid labeled 'A1' containing columns A through E and rows 1 through 7. To the right of the grid is a 'Login' dialog box with fields for 'Agent Name' (containing 'agent') and 'Password' (masked with 'xxxxxxx'). The dialog also features an 'OK' button (highlighted with a red box), 'Cancel', and 'Help' buttons. A small image of an airplane is visible in the background of the dialog.