



ISTQB初级认证

第6章 软件测试工具

作者：郑文强

Email: zwqwwuy@163.com

博客: http://blog.csdn.net/Wenqiang_Zheng

声明

→ 本课件的开发基于**ISTQB Foundation Level Syllabus (Version 2007)**。

→ 感谢**ISTQB**和大纲作者的努力，对应的大纲可以从 www.istqb.org 下载获得。

→ 本课件为个人开发，只能用于个人学习目的，不能用于任何商业活动。

→ 更多**ISTQB**初级认证资料，参考：
http://blog.csdn.net/Wenqiang_Zheng/archive/2011/04/09/6311523.aspx

课程内容

- 1. 测试工具的类型**
- 2. 有效使用工具**
- 3. 组织中工具的引入**

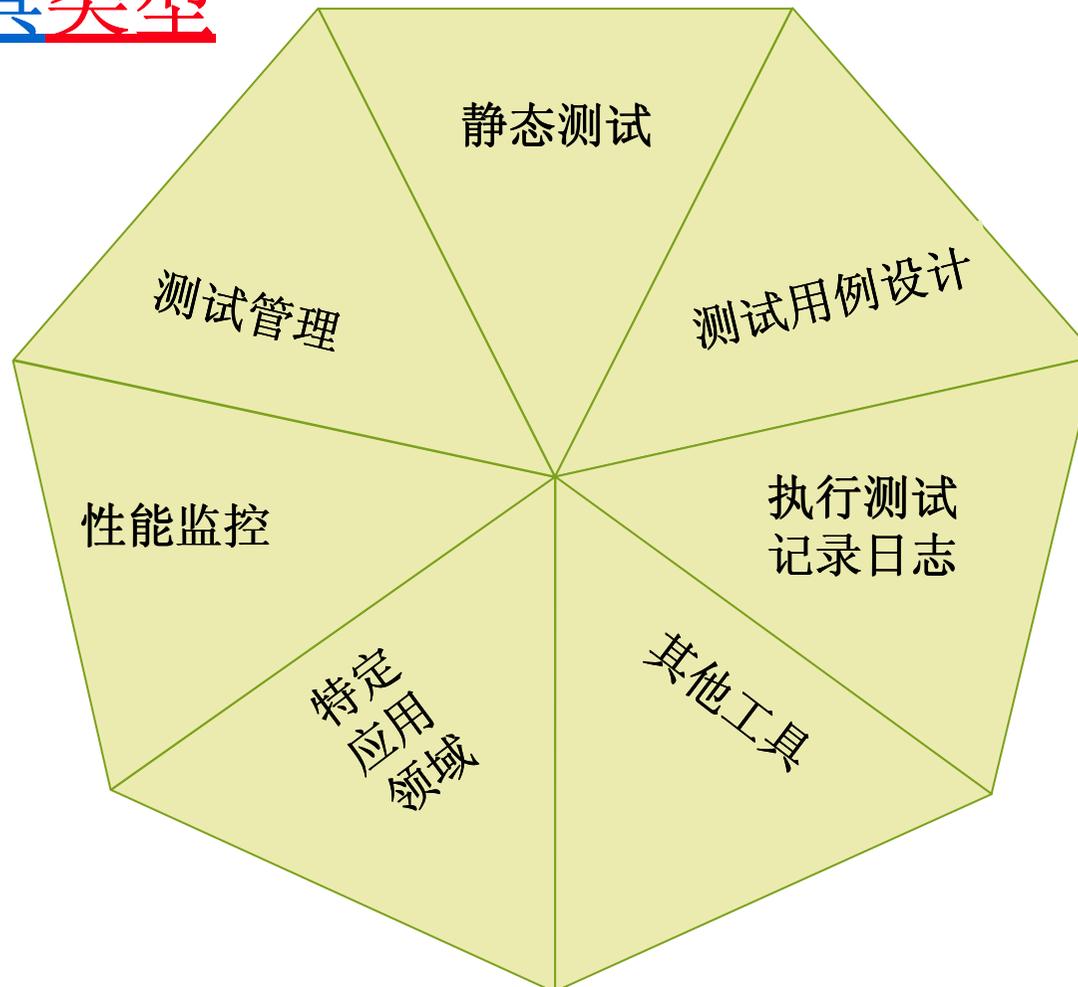
测试工具的类型

ISTQB考试知识点

- ★ 根据测试过程活动，对不同类型的测试工具进行分类（**K2**）；
- ★ 了解能够帮助开发者进行测试的工具（**K1**）；

测试工具的类型

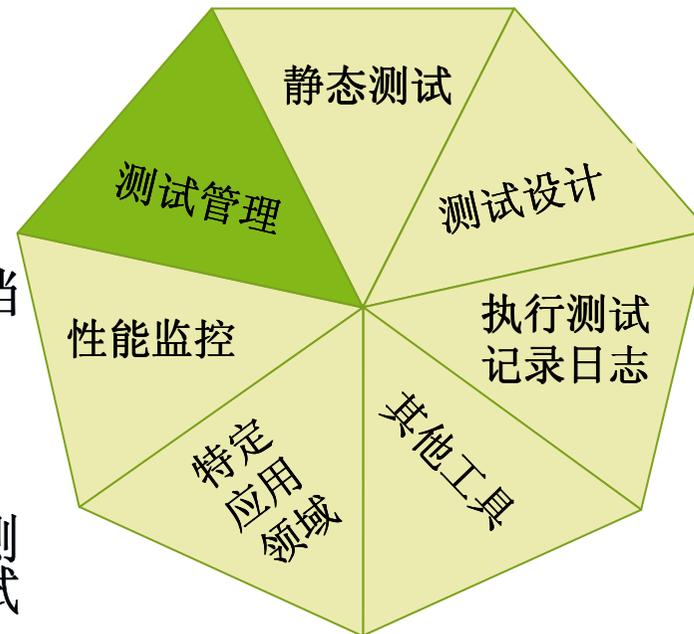
测试工具类型



测试管理的工具支持

测试管理工具的特点

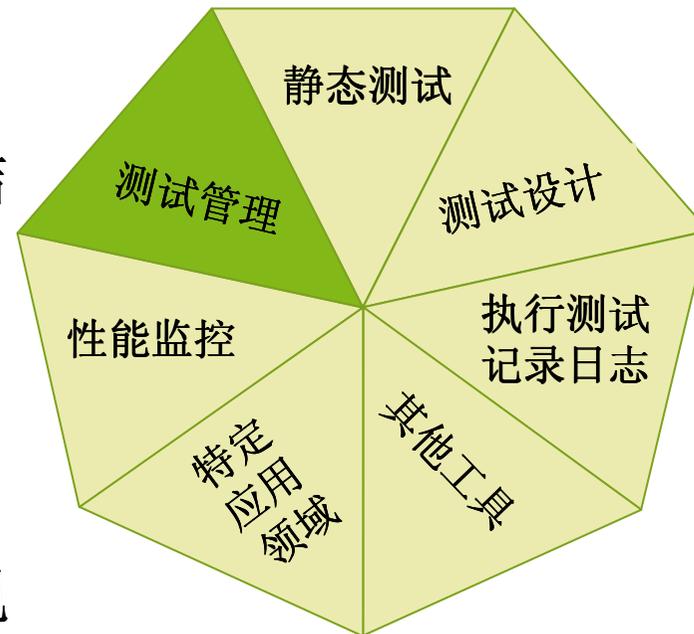
- ★ 适用于整个生命周期的所有测试活动；
- ★ 支持测试管理和测试活动的执行；
- ★ 支持从测试、测试结果、事件到源文档（如需求规格说明）之间的可追溯性；
- ★ 记录测试结果并生成进度报告；
- ★ 对测试以及测试对象，如提交的事件，进行定量分析和度量，以便提供关于测试对象的信息，并帮助控制和改善测试过程和测试质量；
- ★ 提供的接口：
 - ✦ 事件管理；
 - ✦ 配置管理；
 - ✦ 需求管理；
 - ✦ 测试执行工具；



测试管理的工具支持

需求管理工具

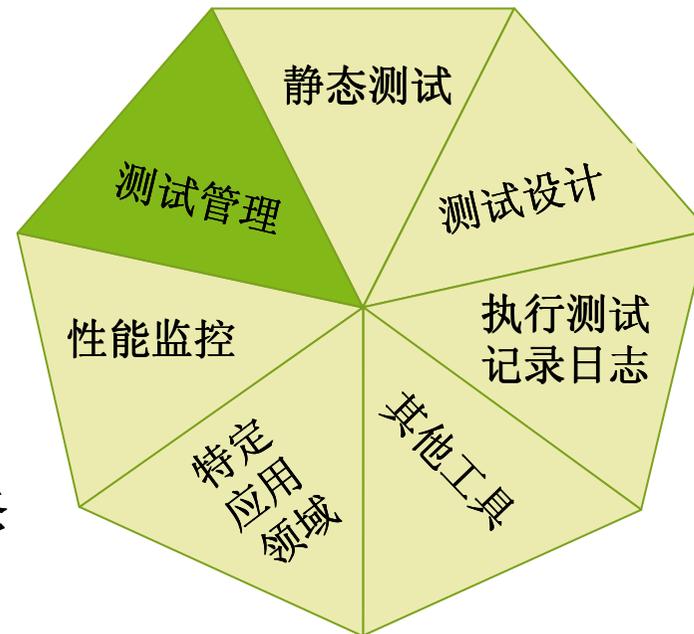
- ★ 储存了需求描述、一致性检查结果和未定义或者丢失的需求；
- ★ 允许对需求设定不同的优先级；
- ★ 测试和需求、功能和/或特征等进行关联；
- ★ 可以在测试管理进度报告中体现可需求的追溯性，以及测试对需求、功能和/或特性的覆盖率；



测试管理的工具支持

事件管理工具

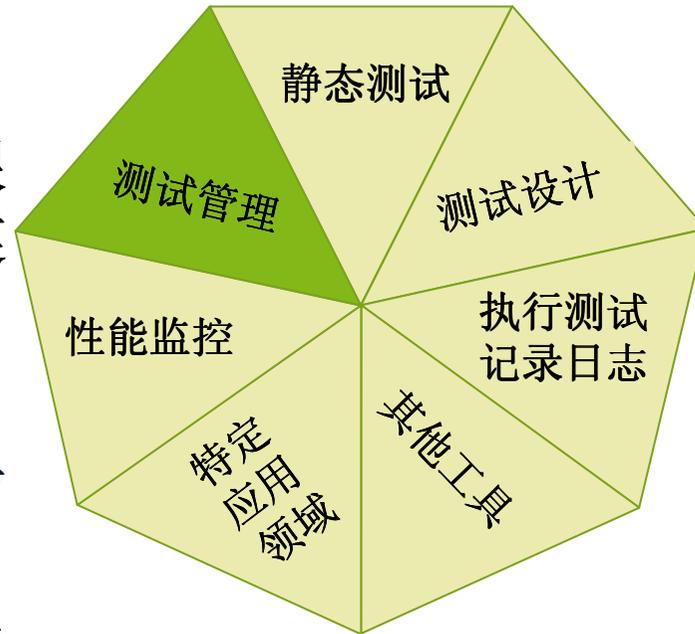
- ★ 储存和管理事件报告：即缺陷、失效、或察觉到的问题和异常等；
- ★ 便于划分事件优先级；
- ★ 给相应的人员分配任务，比如修改或进行确认测试；
- ★ 状态的属性，比如拒绝、待测、延期到下个版本等；
- ★ 不断监测事件的进展，支持统计分析并提供事件的统计报告。它们也称为缺陷跟踪工具；



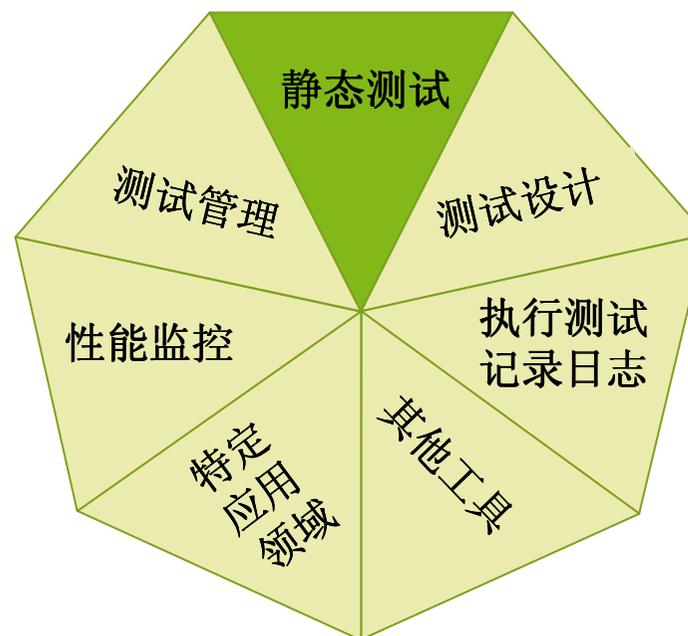
测试管理的工具支持

配置管理工具

- ★ 不是直接的测试工具，但它对跟踪不同软件和测试的版本和构建（**build**）是非常必要的；
- ★ 配置管理工具特点：
 - ✦ 储存软件和测试件的版本和构建信息；
 - ✦ 实现测试件、软件工作产品、产品变更之间的可追溯性；
 - ✦ 特别适用于开发多个硬/软件环境配置的情况（比如不同的操作系统版本、不同的库或编译器，不同的浏览器或不同的计算机）；



静态测试的工具支持

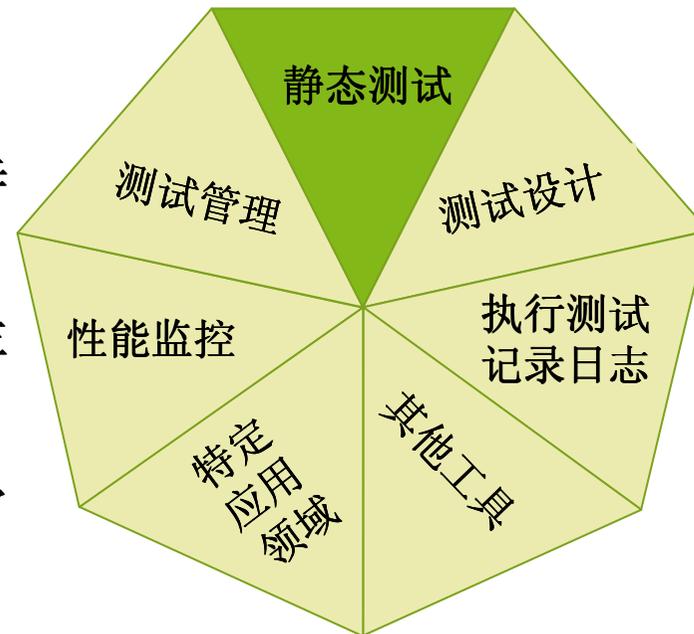


静态分析工具能够尽早的发现缺陷，有助于经济有效的修改缺陷，并不断的改进开发和测试过程！

静态测试的工具支持

评审工具

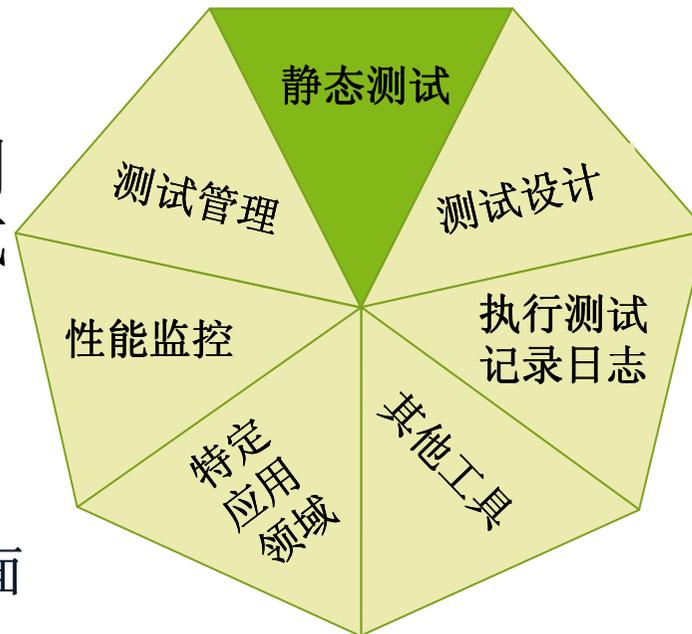
- ★ 评审工具，也称为评审过程支持工具；
- ★ 可以储存评审过程的信息，保存和交流评审意见；
- ★ 报告缺陷和工作量，管理对评审规则和/或检查表的引用；
- ★ 跟踪文档和源代码之间的可追溯性；
- ★ 也可以提供在线评审，这对于团队在不同地区的情况是非常有用的；



静态测试的工具支持

静态分析工具 (D)

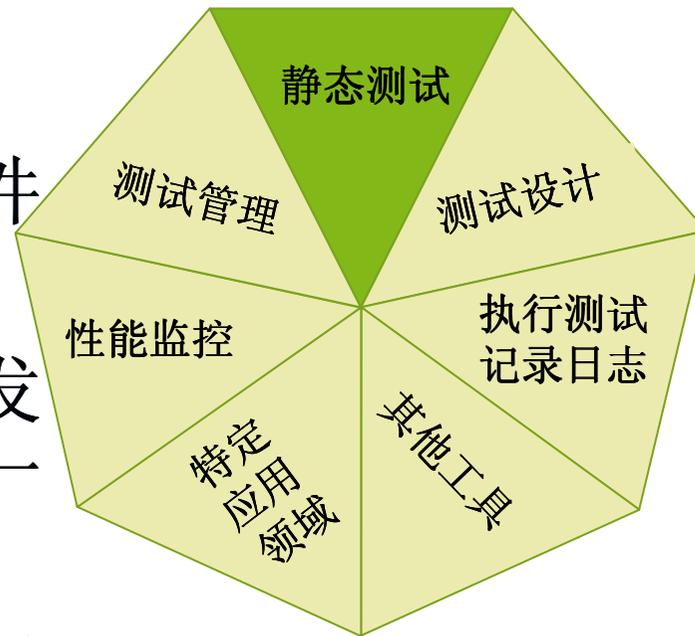
- ★ 静态分析工具支持开发人员、测试员和质量保证人员在动态测试之前发现缺陷；
- ★ 静态分析工具主要的目：
 - ✦ 代码标准的强制性执行；
 - ✦ 结构和依赖性分析，比如**WEB**页面之间的链接；
 - ✦ 帮助分析代码；
- ★ 静态分析工具可以根据代码计算度量值（比如复杂度），从而可以为计划和风险分析等提供有价值的信息；



静态测试的工具支持

建模工具 (D)

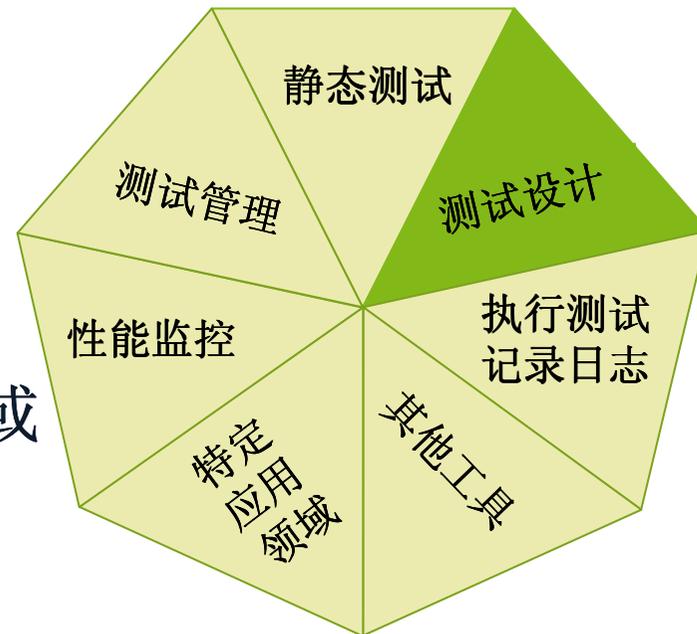
- ★ 建模工具可以用来确认软件模型；
- ★ 数据库模型检查程序可以发现数据模型中的缺陷和不一致；
- ★ 可以在状态模型或对象模型中发现缺陷；
- ★ 可以帮助设计一些基于模型的测试用例；



测试用例设计的工具支持

测试设计工具

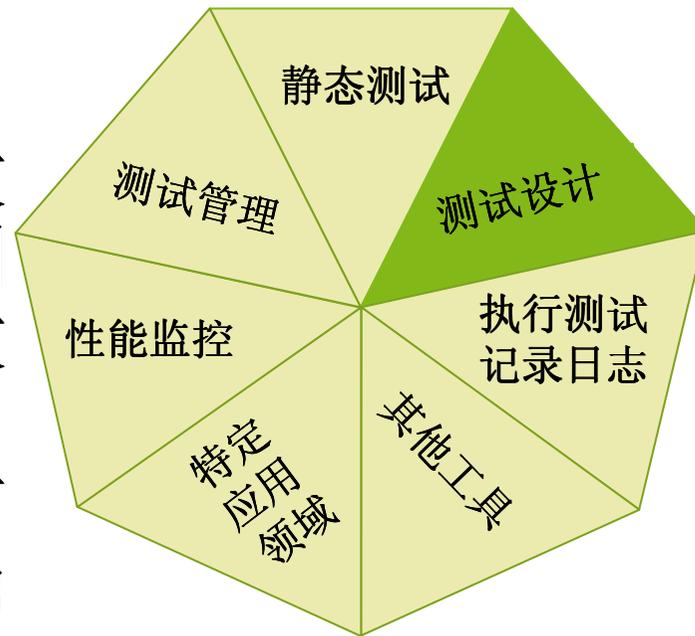
- ★ 测试设计工具的输入：
 - ✦ 正式的需求文档；
 - ✦ 图形化用户接口；
 - ✦ 设计模型，比如状态、数据或对象；
 - ✦ 根据代码；
- ★ 产生测试预期结果；
- ★ 提供测试覆盖率；



测试用例设计的工具支持

测试数据准备工具

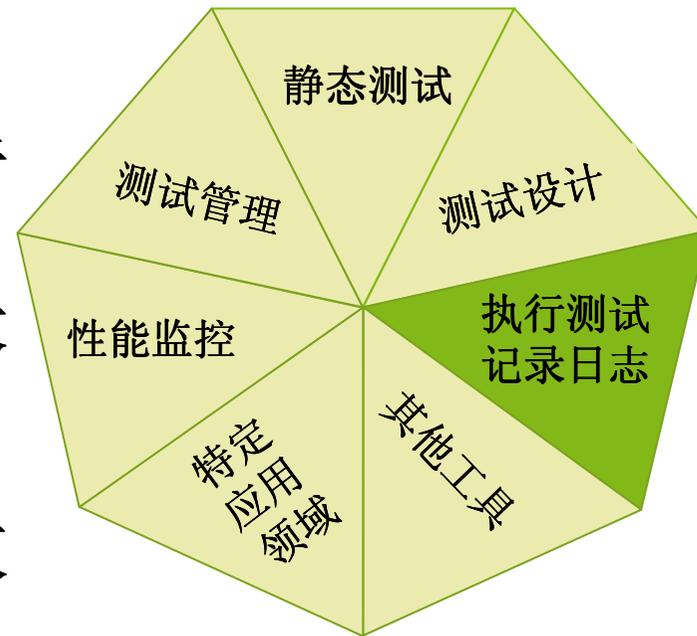
- ★ 用来处理数据库、文件或数据传输，并且生成可以在测试执行过程中使用的测试数据；
- ★ 这种类型的工具的优点是可以保证传输到测试环境中的实时数据是匿名的，从而提供数据保护；



测试执行和记录的工具支持

测试执行工具

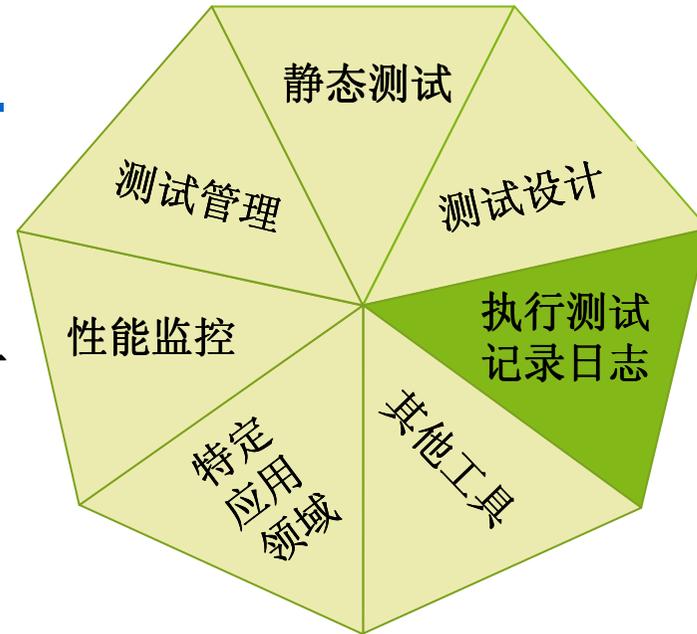
- ★ 自动化或半自动化测试执行用例；
- ★ 通过脚本语言的方法，记录用户操作（也称“录制 / 回放工具”）；
- ★ 针对记录的测试脚本，修改脚本；
- ★ 检查测试结果，生成执行日志；
- ★ 提供了重复执行测试用例的可能性；



测试执行和记录的工具支持

测试用具 / 组件测试框架工具 (D)

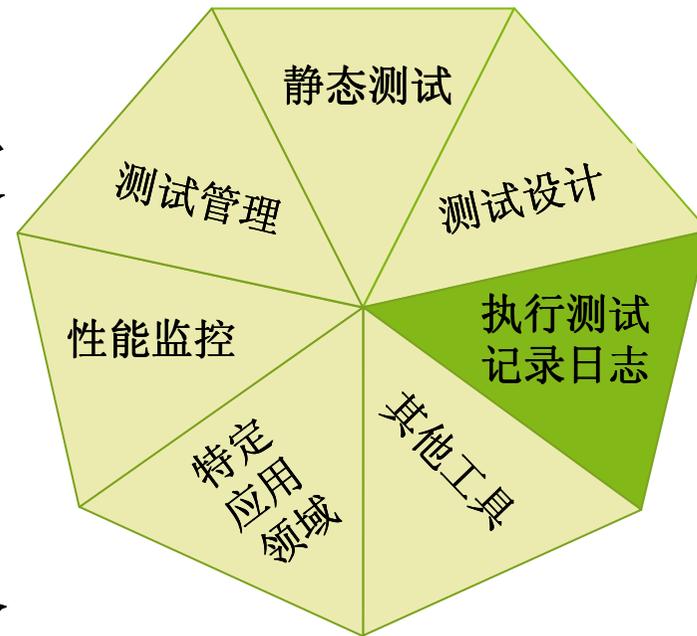
- ★ 提供测试对象的测试环境，模拟系统没有完成或不能呈现的部分；
- ★ 使用测试桩和驱动程序；
- ★ 也称为组件测试结构工具，通常开发人员使用；



测试执行和记录的工具支持

对照工具 (D)

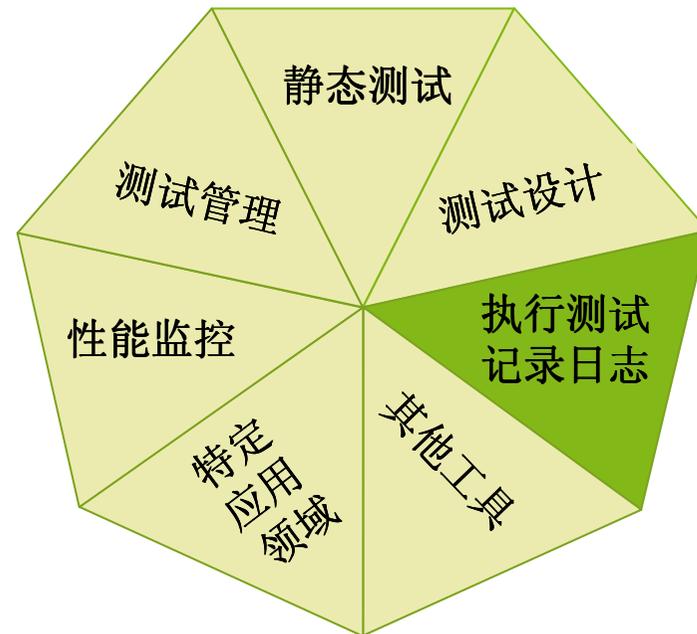
- ★ 测试执行之前和之后，确定不同之处：
 - ✦ 文档；
 - ✦ 数据库；
 - ✦ 测试结果；
- ★ 通常为测试执行工具的一个组件，但是也存在独立的工具；



测试执行和记录的工具支持

覆盖率测量工具 (D)

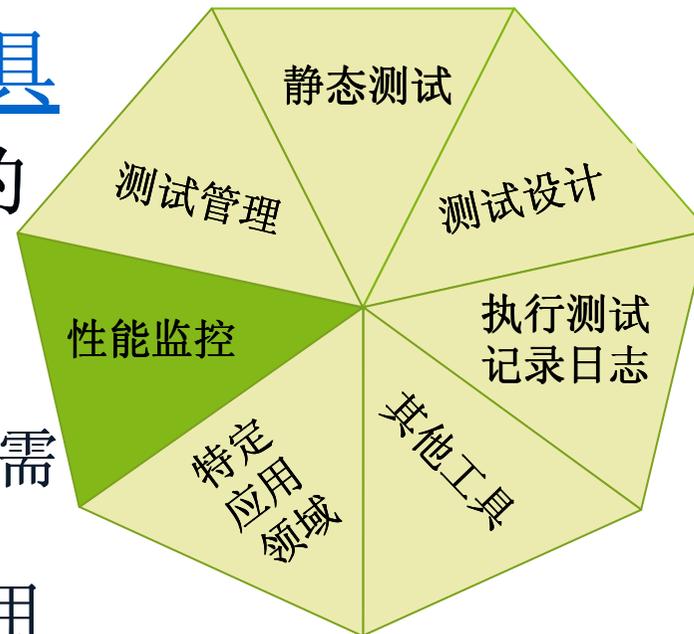
- ★ 提供度量，比如需求覆盖率、分支覆盖率、条件覆盖率等的功能；
- ★ 很有可能会有出入，如改变测试对象；



性能监控的工具支持

性能、负载和压力测试工具

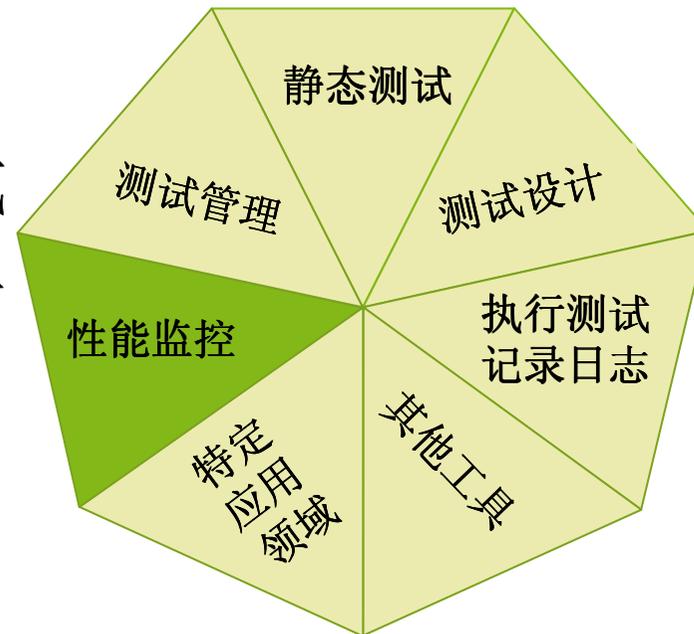
- ★ 短时间内或同时执行大量的测试用例；
- ★ 给以下问题提供了答案：
 - ✦ 系统处理一个正常事务最短需要多少相应时间算正常？
 - ✦ 系统处理最大负责（数据、用户、事务）需要多少相应时间/错误发生率？
 - ✦ 如果超过系统最大负载能力，将如何反应？系统什么时候崩溃？



性能监控的工具支持

监控工具

- ★ 狭隘地说，并非真正的测试工具。但是监控工具提供显示了测试工具的执行结果；
- ★ 记录了特定系统资源的使用情况；
- ★ 通过存储测试的系统和本，确保结果的可溯性；

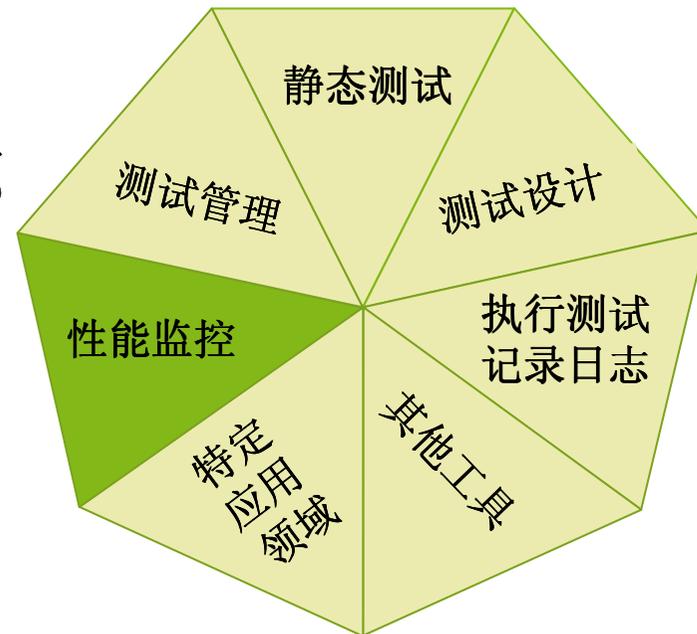


性能监控的工具支持

动态分析工具

★ 在测试执行过程中只发现那些会引起失效的错误：

- ◆ 时间；
- ◆ 内存瓶颈等等；



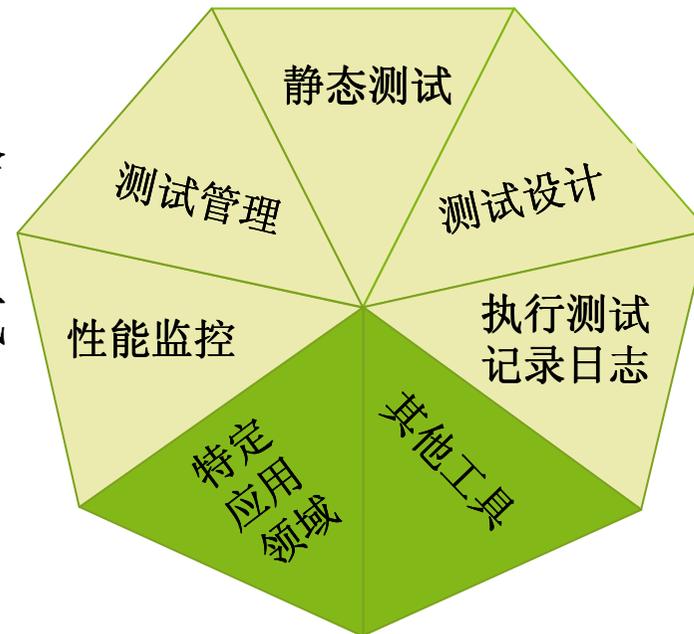
支持测试的其他工具

特定应用领域

- ★ 在测试执行过程中只发现那些会引起失效的错误;
- ★ 网页应用程序的特殊的性能测试工具;
- ★ 网页链接检查器;
- ★ 嵌入式系统的测试工具等等;

其他领域的工具支持

- ★ 电子表格;
- ★ **SQL**;
- ★ 调试等等;

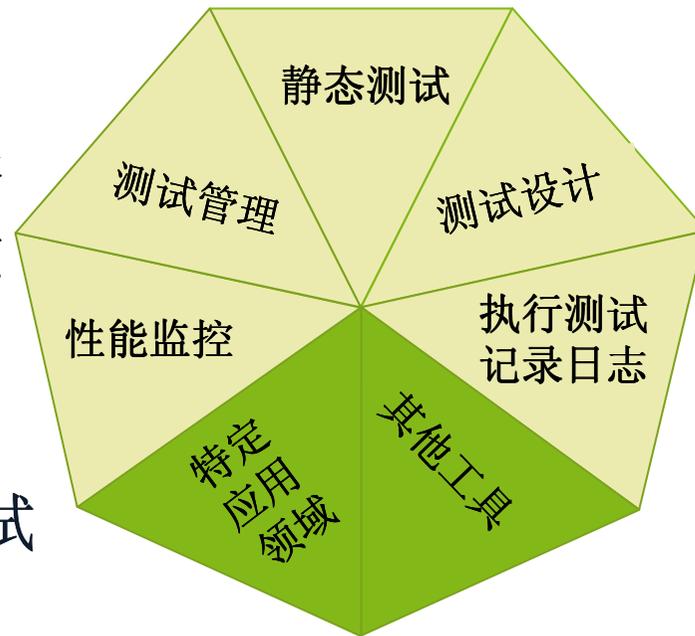


支持测试的其他工具

干预工具

★ 有些工具涉入到系统中检查系统的反应，这样的工具称为干预工具(**intrusive tools**)；例如：

- ★ 代码中插入度量点的性能测试工具 (= 程序指令)；
- ★ 代码中插入计数器的代码覆盖率的工具；
- ★ 代码中插入输出指令的编写日志出口的工具；



课程内容

- 1. 测试工具的类型**
- 2. 有效使用工具**
- 3. 组织中工具的引入**

有效使用工具

ISTQB考试知识点

- ★ 总结测试自动化和使用测试工具的潜在利益和风险（**K2**）；
- ★ 了解测试执行工具可以有包括数据驱动和关键字驱动的不同脚本技术（**K1**）；

有效使用工具

测试自动化的潜在收益

- ★ 减少重复性的工作（比如，执行回归测试，重新输入相同测试数据，按代码标准检查）。
- ★ 更好的一致性和可重复性（比如，用工具执行测试，从需求导出测试）。
- ★ 客观的评估（比如，静态测量、覆盖率）。
- ★ 容易得到测试和测试的相关信息（比如，关于测试进展的统计和图表，事件发生率和性能）。

有效使用工具

测试自动化的风险

- ★ 对工具存在不切实际的期望（包括工具的功能性和易用性）。
- ★ 低估首次引入工具所需的时间、成本和工作量（包括培训和获取外部的咨询）。
- ★ 低估从工具中获得较大和长久收益需要付出的时间和工作量（包括更改测试过程并不断改进工具使用方式的需要）。
- ★ 低估对测试工具生成的结果进行维护所需的工作量。
- ★ 对测试工具过分依赖（替代测试设计或者对一些更适合手工测试的方面使用测试工具）。

有效使用工具

数字驱动

- ★ 数据驱动的方法是将测试输入（测试数据）与测试用例分离，并将测试输入存放在一个电子表格中，这样可以使用不同的数据进行相同的测试。不熟悉脚本语言的测试员可以从一个表格内输入测试数据并执行事先定义好的测试脚本。

有效使用工具

关键字驱动

- ★ 在关键字驱动的方法中，电子表格含有描述系统要采取的行为的关键字（也称为行为字(**action words**)）和测试数据。即使测试员不熟悉脚本语言也能用定义好的关键字来定义测试，而这些关键字又可以针对被测应用程序定制和适配。

课程内容

- 1. 测试工具的类型**
- 2. 有效使用工具**
- 3. 组织中工具的引入**

组织中工具的引入

ISTQB考试知识点

- ★ 阐述将工具引入组织中的主要步骤（**K1**）；
- ★ 阐述为评估工具所进行的学习调查/试点项目阶段的目的（**K1**）；
- ★ 了解要获得好的工具支持，仅靠购置工具是不够的，还需要考虑其他因素（**K1**）；

组织中工具的引入

为组织选择一个工具所需要考虑的关键点有：

- ★ 评估组织的成熟度(**maturity**)、分析引入工具的优点和缺点和认识引入工具能改善测试过程的可能性。
- ★ 根据清晰的需求和客观的准则进行评估。
- ★ 以概念验证(**proof-of-concept**)来检验工具应该具有的功能，决定工具是否满足要求。
- ★ 对工具提供商进行评估（包括培训、提供的支持及其他商业方面）。
- ★ 确定在工具使用方面应提供的指导和内部培训需求。

组织中工具的引入

将选择的工具引入组织要从一个试点项目开始，试点项目有下面的目的：

- ★ 对工具有更多的认识。
- ★ 评价工具与现存的过程以及实践的配合程度，确定哪些方面需要作修改。
- ★ 决定工具和由工具生成/应用的结果的使用、管理、保存和维护的使用工具标准（比如，文件和测试的命名规则、创建数据库和定义测试套件 (**test suites**)）。
- ★ 评估在付出合理的成本后能否得到收益。

组织中工具的引入

在组织内成功部署工具的因素包括：

- ★ 逐步在组织的其他部门推广工具。
- ★ 调整并改进过程来配合工具的使用。
- ★ 为新使用者提供培训和指导。
- ★ 定义使用指南。
- ★ 找到并实施学习工具使用方面教训的方法。
- ★ 监测工具的使用和收益情况。

答疑

