

认证测试工程师 测试自动化策略 教学大纲

版本: EN1.0_CN1.0

发布日期: 2025 年 7 月 1 日

国际软件测试认证委员会



中文版的翻译、编辑和出版统一由 ISTQB® 授权的 CSTQB® 负责



若您对此文档有任何问题, 欢迎您扫码添加【官方微信号】反馈。

版权声明

英文版权声明

版权声明©国际软件测试认证委员会（以下简称 ISTQB®）

ISTQB®是国际软件测试认证委员会的注册商标。

2024 年版©测试自动化策略 v1.0 教学大纲的作者：Andrew Pollner（主席）、Péter Földházi、Patrick Quilter、Gergely Ágnesz、László Szikszai

版权所有。作者特此将版权转让给 ISTQB®。作者（作为当前版权持有人）和 ISTQB®（作为未来版权持有人）已同意以下使用条款：

作为非商业化用途的摘录和复制，在来源得到确认的情况下是可以的。对于培训机构，认可此大纲的作者和 ISTQB®是此大纲的来源和版权所有者，并且此类培训课程的任何广告只有在收到 ISTQB®认可的成员国委员会（中国是 CSTQB®）对培训材料的正式认证后才能提及大纲，任何经认证的培训机构都可以使用本大纲作为培训课程的基础。

如果认可作者和 ISTQB® 是此大纲的来源和版权所有者，任何个人或团体都可以使用本大纲作为文章和书籍的基础。

未经 ISTQB®书面批准，禁止以其他方式使用本大纲。

ISTQB®认可的任何成员国委员会都可以翻译本大纲，前提是他们在教学大纲的翻译版本中复制上述版权声明。

中文版权声明

未经许可，不得复制或抄录本中文版文档内容。

版权标志©国际软件测试认证委员会中国分会（以下简称“CSTQB®”）

修订记录

版本	日期	备注
教学大纲 V1.0	2024/05/03	CT-TAS V1.0 GA 正式发布
EN1.0_CN1.0	2025/07/01	CT-TAS v1.0 大纲中文本地化发布

目录

版权声明	1
修订记录	2
目录	3
致谢	5
0. 简介	6
0.1 本大纲的目的	6
0.2 软件测试中的测试自动化策略	6
0.3 测试人员和测试自动化工程师的职业道路	6
0.4 商业价值	7
0.7 课程认证	8
0.9 保持时效性	9
0.10 详细级别	9
1. 测试自动化策略的简介和目标-45 分钟（K2）	11
1.1 测试自动化项目的成功要素	12
1.1.1 定义测试自动化策略的目的和目标	12
1.1.2 识别测试自动化项目成功的技术因素	12
1.1.3 总结选择测试自动化候选项目进行投资的适用准则	14
2. 测试自动化资源-60 分钟（K2）	15
2.1 实现测试自动化解决方案的成本和风险	16
2.1.1 在所有权成本方面比较替代技术方案	16
2.1.2 解释测试自动化工具软件许可模式的考虑事项	16
2.1.3 提供定义测试自动化策略时要考虑要素的示例	17
2.2 测试自动化中的角色和职责	18
2.2.1 总结成功的测试自动化方案所需的角色和技能	18
3. 准备测试自动化-225 分钟（K3）	20
3.1 跨测试级别的集成	21
3.1.1 区分测试自动化的不同分布情况	21
3.1.2 根据被测系统架构选择测试自动化方法	22
3.1.3 展示优化测试自动化的分布，以实现左移和右移方法	23
3.2 不同软件开发生命周期模型中的策略考虑	24
3.2.1 解释测试自动化项目如何与传统的软件开发生命周期模型相符合	24
3.2.2 解释测试自动化项目如何与支持测试自动化的敏捷软件开发最佳实践相符合	24
3.2.3 为测试自动化项目做好准备，使其符合支持持续测试中测试自动化的 DevOps 最佳实践	24
3.3 测试自动化的应用和可行性	25
3.3.1 解释用于确定测试是否适合进行测试自动化的准则	25
3.3.2 识别只有测试自动化能解决的挑战	25
3.3.3 识别难以自动化的测试条件	26
4. 测试自动化的组织部署与发布策略-135 分钟（K2）	27
4.1 测试自动化解决方案策划	28
4.1.1 识别测试自动化如何支持缩短上市时间的方式	28
4.1.2 识别测试自动化如何帮助根据需求验证已报告的缺陷的方式	28
4.1.3 定义为运营相关的场景开发测试自动化的方法	29
4.2 测试自动化部署策略	30

4.2.1	定义测试自动化的部署策略.....	30
4.2.2	识别测试自动化在部署中的风险.....	31
4.2.3	定义降低测试部署风险的方法.....	32
4.3	测试环境中的依赖关系	33
4.3.1	定义测试环境中的测试自动化组件.....	33
4.3.2	识别测试自动化的基础组件及其依赖关系.....	33
4.3.3	为了与被测系统进行集成定义测试自动化的数据和接口需求.....	34
5.	测试自动化影响分析-150 分钟（K3）	35
5.1	建立和维护测试自动化的投资.....	36
5.1.1	展示构建测试自动化解决方案的投资回报.....	36
5.2	测试自动化度量.....	37
5.2.1	对测试自动化度量进行分类.....	37
5.3	测试自动化在项目和组织层面的价值.....	39
5.3.1	确定使用测试自动化的组织级考虑因素.....	39
5.3.2	分析有助于确定测试自动化之测试目标的最佳实施的项目特征.....	40
5.4	根据测试自动化报告做出决策.....	41
5.4.1	分析测试报告数据引导决策制定.....	41
6.	测试自动化的实现和改进策略 - 150 分钟（K3）	43
6.1	从手动测试到持续测试的迁移活动.....	44
6.1.1	描述从手动测试迁移到测试自动化的要素及策划活动.....	44
6.1.2	描述从测试自动化迁移到持续测试的要素及策划活动.....	45
6.2	整个组织的测试自动化策略.....	46
6.2.1	进行测试自动化资产和实践的评估，以确定改进领域.....	46
7.	参考文献	48
8.	附录 A - 学习目标/知识认知级别.....	51
9.	附录 B 业务成果与学习目标的追溯矩阵.....	53
10.	附录 C - 版本说明	57
11.	附录 D - 领域专用术语.....	58
12.	索引	59

致谢

本大纲由 ISTQB®成员国大会于 2024 年 5 月 3 日正式发布。

本大纲由国际软件测试认证委员会的专家工作组的测试自动化任务组制作：Graham Bath（专家工作组主席）Andrew Pollner（专家工作组副主席兼测试自动化任务组主席），Péter Földházi, Patrick Quilter, Gergely Ágnes, László Szikszai。测试自动化任务组的评审人员包括：Armin Beer, Armin Born, Geza Bujdosó, Renzo Cerquozzi, Jan Giesen, Arnika Hryszko, Kari Kakkonen, Gary Mogyoródi, Chris van Bael, Carsten Weise, Marc-Florian Wendland。

技术评审：Gary Mogyoródi

以下人员参与了本教学大纲的评审、评论和投票：

Horváth Ágota, Laura Albert, Prasunkumar Banerjee, Jürgen Beniermann, Armin Born, Piet de Roo, Nicola De Rosa, Dingguofu Ding Guofu, Elizabeta Fournieret, Jan Giesen, Erik Haartmans, Matthias Hamburg, Tobias Horn, Mattijs Kemmink, Ilia Kulakov, Ashish Kumar, Vincenzo Marrazzo, Marton Matyas, Patricia McQuaid, Smitha Mohandas, Ingvar Nordström, Sreeja Padmakumari, Nishan Portoyan, Meile Posthuma, Swapnil Shah, Péter Sótér, Szilard Szell, Richard Taylor, Giancarlo Tomasig, Chris Van Bael, Daniel Van der Zwan, Carsten Weise, Marc-Florian Wendland, Claude Zhang。

ISTQB® CT-TAS 专业-测试自动化策略大纲 V1.0 文档中文翻译、评审及 QA 评审参与者（按姓氏拼音排序）：陈耿（QA 评审）、龚雄、刘晓玲、吴洁、左振雷。

0. 简介

0.1 本大纲的目的

本大纲是国际软件测试认证委员会测试自动化策略专家认证的的基础。ISTQB®提供本大纲的目的如下：

1. 各成员国认证委员会翻译成本地语言并授权给培训机构。成员国委员会可以根据特定的语言调整大纲，并修改参考文献列表以适应本地出版物。
2. 认证机构使用当地语言编写适合本大纲学习目标的考试题。
3. 培训机构编制课件并确定适当的教学方法。
4. 认证考试的考生准备认证考试（认证考试可以作为培训课程的一部分或独立进行）。
5. 国际软件和系统工程界以此推动软件和系统测试专业的发展，并作为著书和写文章的基础。

0.2 软件测试中的测试自动化策略

测试自动化策略专家认证针对所有软件测试和测试自动化的人员，包括测试员、测试分析师、测试自动化工程师、测试顾问、测试架构师、测试经理和软件开发人员等角色。这个专家认证也适用于任何想要对测试自动化有基本了解的人，比如项目经理、质量经理、软件开发经理、业务分析师、IT 主管和管理顾问。

专家级测试自动化策略（CT-TAS）教学大纲提出了在组织内进行测试自动化策划时发挥作用的多个因素。技术工程实施相关的测试自动化方法和最佳实践的内容不在本教学大纲范围内，而是包含在单独的 ISTQB CTAL-TAE 教学大纲中。

测试自动化策略涉及了超越技术工具的实现和集成挑战的更广泛的测试自动化需求。测试自动化的策略以系统化和一致的方式提供了组织内跨项目实现的愿景，最终向组织展示了其价值。

0.3 测试人员和测试自动化工程师的职业道路

ISTQB®计划为处于职业生涯各个阶段的测试专业人员提供支持，提供广泛和深入的知识。获得 ISTQB®测试自动化策略专家认证的个人也可能对测试自动化工程师（CTAL-TAE）认证感兴趣。

获得 ISTQB®认证测试工程师-测试自动化策略专家认证的个人也可能对核心高级（测试分析师、技术测试分析师和测试经理）感兴趣，和之后的专家级（测试管理或改进测试过程）感兴趣。任何在

敏捷环境领域寻求发展测试实践技能的人，都可以考虑敏捷技术测试人员或大规模敏捷测试领导力认证。专家级模块深入研究了特定测试方法和测试活动领域，例如测试自动化工程、性能测试、安全测试、AI 测试和移动应用程序测试，或者需要领域专有技术的场合（例如汽车软件测试或游戏测试）。有关 ISTQB 认证测试人员计划的最新信息，请访问 www.istqb.org。

0.4 商业价值

本节列出了获得测试自动化策略专家认证的测试人员所预期的商业价值。

经过测试自动化策略专家认证的测试人员可以…

TAS-B01	理解软件和系统中影响测试自动化成功的因素。
TAS-B02	识别实施测试自动化解决方案的成本和风险。
TAS-B03	理解对测试自动化做出贡献人员的角色和职责。
TAS-B04	跨测试级别的测试自动化集成计划。
TAS-B05	识别在不同软件开发生命周期模型中实施测试自动化的战略考量因素。
TAS-B06	理解测试自动化的适用性和可行性。
TAS-B07	策划满足组织需求的测试自动化解决方案。
TAS-B08	理解测试自动化部署策略。
TAS-B09	理解测试环境中测试自动化的依赖关系。
TAS-B10	理解建立和维护测试自动化的成本。
TAS-B11	了解哪些测试自动化指标有助于推动决策制定。
TAS-B12	识别测试自动化为项目和组织带来价值的途径。
TAS-B13	识别测试自动化报告的需求，以满足利益相关人的需要。
TAS-B14	定义从手动测试到测试自动化的转换活动。
TAS-B15	制定测试自动化策略，来确保项目之间共享资产和方法，从而在整个组织内实现一致性的实施。

0.5 可考核的学习目标与知识认知级别

学习目标支持商业价值，并用于创建测试自动化策略专家测试人员的认证考试题。

一般来说，除大纲简介和附录外，本大纲的所有内容均可在 K2 和 K3 级进行考核。也就是说，考生可能会被要求识别（Recognize）、理解（Remember）或回顾（Recall）这六个章中提到的关键词或概念。具体的学习目标级别会显示在每章的开头部分，并按照以下分类：

- K2: 理解
- K3: 应用

附录 A 中给出了学习目标的详细内容和示例。

即使学习目标中没有明确提及，所有列在章节标题下方作为关键词的术语也都应牢记。

0.6 测试自动化策略专家认证考试

测试自动化策略专家认证考试将基于本教学大纲。考试题的答案可能需要使用基于本教学大纲多个部分的内容。除大纲简介和附录之外，大纲的所有部分都可以进行考核。标准和书籍作为参考资料，但除了大纲本身所摘要的标准和书籍内容外，其他内容不会被考核。

有关测试自动化策略教学大纲认证考试的更多详细信息，请参阅与基础级和高级以及专家级教学大纲模块兼容的考试结构和规则 v1.1 文档。

参加测试自动化策略认证考试的入门标准是考生对软件测试和测试自动化感兴趣。但是，强烈建议考生还应：

- 至少具备软件和系统开发的最低背景，以及在企业中技术实施的领导能力，以及作为高级测试工程师、测试主管或软件开发人员的经验。
- 参加一门由 ISTQB 标准授权的课程（由 ISTQB 认可的成员委员会之一认证）。

基本要求事项：在参加测试自动化策略教学大纲专家认证考试之前，应获得 ISTQB®基础级证书。

0.7 课程认证

ISTQB®成员国委员会可以对培训机构的课程材料是否遵循本教学大纲进行认证。培训机构应从委员会或执行认证的机构获取认证指南。经认证的课程被认为符合本教学大纲，并且可以将 ISTQB® 考试作为课程的一部分。

本教学大纲的认证指南遵循过程管理与合规工作组发布的通用认证指南。

0.8 标准的处理

测试自动化策略教学大纲中引用了一些标准（例如 IEEE、ISO 等）。这些参考文献的目的是提供框架（例如有关质量特性的 ISO 25010 参考文献），或者提供附加信息来源（如果读者需要）。请注意，教学大纲使用这些标准作为参考。标准文件不作为本大纲的考核内容。有关标准的更多信息，请参阅第 7 章。

0.9 保持时效性

软件行业变化迅速，为了应对这些变化并让利益相关方能够获取最新的相关信息，ISTQB®工作组在 www.istqb.org 网站上创建了链接，其中有些链接涉及支持文档和标准的变化。这些信息不作为测试自动化策略专家教学大纲考核内容。

0.10 详细级别

本教学大纲的详细程度为有助于国际课程和考试保持一致。为了达到这一目标，本大纲由下面几个部分组成：

- 描述测试自动化策略专家意图的一般教学目标。
- 列出了学员必须能回顾的术语（关键词）列表。
- 每个知识领域的学习目标，描述需要达到的认知学习成果。
- 对重要概念的描述，包括来源参考，例如：公认的文献或标准。

本教学大纲内容并不是对软件测试的整个知识领域的描述，仅提供了测试自动化策略专家培训课程所应涵盖的详细程度。它侧重于可以所有软件项目都能应用到的测试概念和技术，包括那些遵循敏捷方法的项目。本教学大纲不包含任何与敏捷测试相关的特定学习目标，但确实讨论了如何在敏捷项目和其他类型的项目中应用这些概念。

0.11 大纲的结构

本大纲共有六章考核内容。每章的一级标题指定该章的培训时长。每章的各节标题不再提供时长安排。对于已认证的培训课程，教学大纲要求至少 12.75 小时的教学时长，分布在以下六个章节中：

- 第 1 章：45 分钟 - 测试自动化策略的简介和目标
 - 理解测试自动化的概念，并学习候选项目的选择准则。
 - 理解定义成功的测试自动化实现的因素。
- 第 2 章：60 分钟 - 测试自动化资源
 - 学习适用于测试自动化的不同解决方案，以及每个解决方案的相对成本。
 - 涵盖测试自动化工具的软件授权。
 - 了解测试自动化所需的技能。

- 第 3 章：225 分钟 - 准备测试自动化
 - 学习如何跨测试级别和在测试级别内运用测试自动化。
 - 涵盖了适当分布测试和实现左移/右移的测试自动化策略。
 - 学习测试自动化如何支持遗留和敏捷项目。
 - 介绍了 DevOps 中的测试自动化和持续测试实践。
 - 理解如何定义使用测试自动化的准则，包括最适合测试自动化的测试。
- 第 4 章：135 分钟 - 测试自动化的组织部署和发布策略
 - 学习测试自动化如何加快上市时间。
 - 了解如何开发运营相关的自动化测试和缺陷报告。
 - 测试自动化的部署策略和风险缓解。
 - 学习测试自动化环境及其依赖关系。
 - 将测试自动化和测试数据集成到被测系统。
- 第 5 章：150 分钟 - 测试自动化影响分析
 - 测试自动化指标和报告，以辅助明智的决策。
 - 学习如何实现测试自动化的投资回报。
 - 组织和项目使用测试自动化的目标。
 - 学习如何分析测试报告，并以清晰易懂的方式报告给决策者。
- 第 6 章：150 分钟 - 实施和改进测试自动化策略
 - 学习如何从手动过渡到测试自动化和持续测试。
 - 评估测试自动化以实现持续改进。

1. 测试自动化策略的简介和目标-45 分钟 (K2)

关键词

测试自动化架构（test automation architecture），测试自动化框架（test automation framework），测试自动化策略（test automation strategy）

第 1 章的学习目标：

1.1 测试自动化项目的成功因素

- CT-TAS-1.1.1 (K2) 解释测试自动化的目标和相关性。
- CT-TAS-1.1.2 (K2) 识别测试自动化项目成功的技术因素。
- CT-TAS-1.1.3 (K2) 总结选择测试自动化候选项目进行投资的适用准则。

1.1 测试自动化项目的成功要素

1.1.1 定义测试自动化策略的目的和目标

在定义测试自动化项目的策略时，需要考虑以下要点：

- 定义预期目的。
- 识别风险。
- 定义范围。
- 识别所涉及的利益相关者。
- 选择用于自动化的工具。
- 设计测试自动化架构（TAA）。
- 识别环境。

测试自动化的目标可能包括：

- 提高测试效率。
- 提供更广泛和更深入的覆盖率。
- 提高被测系统的整体质量。
- 降低总成本并缩短上市时间。
- 执行手动测试人员无法执行的测试。
- 减少测试执行时间。
- 增加测试频率。

由于敏捷软件开发提供了更快的部署周期，并且云应用程序更加广泛，因此开发活动需要更早和更快的得到被测系统质量的反馈。作为这种转变的结果，考虑到测试自动化的性质和测试目标，其在最新的项目中得到更多的关注。

1.1.2 识别测试自动化项目成功的技术因素

以下成功因素适用于关注于影响项目长期成功的测试自动化项目。使用试点项目有助于确定工具和所选技术的可行性。

测试自动化的成功因素包括：

- 被测系统的可测试性。
 - 使得测试自动化能够访问被测系统的接口。
- 定义的测试自动化策略。
 - 策略必须是适用和可定制的；在时间和成本约束内目标是可实现的，并且需要保持更新。
- 测试自动化架构（TAA）。
 - 澄清实现内容和实现方式。
 - 其他信息请参阅国际软件测试认证委员会高级测试自动化工程师（CTAL-TAE）教学大纲第 3.1.1 节。
- 测试自动化框架（TAF）。
 - TAF 易于使用，有良好的文档记录和可维护性，支持以一致的方法实现自动化测试。其他信息请参阅 CTAL-TAE 教学大纲第 3.1.3 节。
 - 测试报告机制已定义并实现。
 - 容易排除故障。
 - 适当的测试环境。
 - 文档化的自动化测试用例。
 - 可追溯到测试用例定义与需求的自动化测试。
 - 易于维护。
 - 保持最新的自动化测试。
 - 清晰的部署计划。
 - 清晰的测试执行计划。
 - 根据需要退役（或移除）测试用例。
 - 有效的异常处理。

在开始测试自动化项目之前，重要的是通过考虑已有的要素和缺失的要素来分析项目成功的机会，牢记所选择的测试方法以及项目周境的风险。并非所有要素都是必需的，而且在实践中，很少能满足所有要素。

1.1.3 总结选择测试自动化候选项目进行投资的适用准则

在项目上建立测试自动化意味着投资，并且在大多数情况下意味着重大成本。这一点应该与项目的性质以及是否应该在项目上使用测试自动化一起考虑。

在引入测试自动化之前，请考虑以下投资准则：

- 导入成本：除了建立测试自动化所产生的工作量以外，项目还需要额外的成本，因为可能需要雇用新的测试自动化工程师（TAE）、购买新的硬件或安排培训。
- 项目软件开发生命周期（SDLC）的当前阶段：最好尽早开始，使得测试自动化可以更快地带来更多价值。
- 项目/软件开发的预期/计划持续时间：对于短期项目，可能没有足够的资源来启动，或者没有足够的剩余时间让测试自动化带来价值。
- 维护成本：在从零开始的情况下，开发和维护一个测试自动化解决方案（TAS）需要花费的时间。

如果在一个项目上引入测试自动化的投资是可以接受的，测试自动化的准备工作就可以开始了。这包括选择正确的测试方法和测试自动化工具。这个过程的主要责任由测试架构师或测试经理的战略角色承担，他们具备必要的测试自动化知识以便做出相关决策。

2. 测试自动化资源-60 分钟（K2）

关键词

测试自动化工程师（test automation engineer），测试自动化解决方案（test automation solution）

第 2 章的学习目标：

2.1 实现测试自动化解决方案的成本和风险

CT-TAS-2.1.1 （K2） 在所有权成本方面比较替代技术方案。

CT-TAS-2.1.2 （K2） 解释测试自动化工具软件许可模式的考虑事项。

CT-TAS-2.1.3 （K2） 提供定义测试自动化策略时要考虑要素的示例。

2.2 测试自动化中的角色和职责

CT-TAS-2.2.1 （K2） 总结成功的测试自动化方案所需的角色和技能。

2.1 实现测试自动化解决方案的成本和风险

2.1.1 在所有权成本方面比较替代技术方案

在软件所有权方面，一种常见的做法是基于开源或商业工具定制一个机构内的解决方案。其他替代做法包括非定制的商业解决方案，或与外包公司签订合同，让测试自动化工程师（TAE）完成该项工作。

在组织内部创建所有测试自动化解决方案（TAS），可以确保所有相关的成本、资源、风险和管理都在组织内部（例如团队成员/开发人员以及实现实际解决方案所需的硬件和软件资源）。一个关键要素是：通过采用这一方法并为这一活动分配时间，无需与供应商或外包公司签订合同即可开发 TAS，因为每个所需的测试自动化工程师（TAE）及其知识是在组织内现成的。然而，为了成功执行这一方法，组织需要有能够驱动测试自动化解决方案（TAS）开发的合适的测试自动化工程师（TAE），不论是雇佣到的或者培训出的。

使用基于供应商的解决方案，软件所有权将在客户和供应商之间共享，具体取决于他们的合同细节。如果在组织中已经有一个测试工具被试用并且满足所有需求，并且测试工具不需要额外的功能，这种方法将更容易采用。组织内的测试人员一旦接受了来自供应商必要的培训后，就能够高效地开展工作，并且通常会指派技术专家（Subject Matter Experts, SME）作为产品负责人在组织内提供支持。这种做法的风险在于，如果需要完成测试工具的任何额外工作（例如修复工具缺陷和其他功能请求），就需要花费时间，并且需要与供应商进行协商，才能按时以正确的方式完成，这可能会影响使用该测试工具的测试人员的工作。

如果组织不想拥有建立团队的所有权，外包是一个推荐的解决方案。在与外包公司合作的情况下，客户不必租用或购买任何实际的硬件、软件，或者招聘具备所需技能的员工。所有的工作和额外的成本都将由外包公司承担，同时外包公司也将拥有工具的所有权。必须在合同中明确可测量的预期和度量，使得进度透明、可见。如果组织由于项目时间短、成本或其他考虑因素，不打算投入精力招聘测试自动化工程师（TAEs），则建议采用这种方法。

2.1.2 解释测试自动化工具软件许可模式的考虑事项

当建立测试自动化时，软件许可授权是要考虑的一个重要方面。下面提到的每个许可模式对可用性和成本因素都有不同的影响（例如，测试执行的成本、建立开发环境的难度）。

软件许可授权模式包括：

- **开源：**在许多情况下，组织使用开源工具来实现他们在测试自动化中的测试目标。这样做的主要原因是使用测试工具没有许可证成本或任何维护费用。许多用户和组织为开源工具的开发做出了贡献，这使得信息收集和获得支持变得更加容易。大多数开源许可证还允许人们在必要时修改工具，甚至重新发布它们，而没有明显的限制。

- **每用户/机器许可证：**商业工具通常按（每）用户或机器授权。就成本而言，当组织知道将在给定项目上长期工作的测试自动化工程师（TAE）的人数时，就可以有效地使用工具。通常，组织订购的许可证越多，他们获得的价格就越优惠。

- **浮动许可证：**这是一种可以在组织中的多人之间在不同时间共享使用的许可证。当有多个 TAE 在不同的机器上工作时会非常有用。许可证的数量是根据并发使用计算的，而不是根据运行测试的 TAE 或特定计算机的总数计算的。

- **运行时许可证：**许多商业工具都有用于执行测试自动化的运行时许可证。这种类型的授权发生在托管测试自动化工具、多个操作系统和浏览器版本作为服务的云供应商这里。还有一些云供应商提供对设备资源池的访问，这些设备资源池具备各种移动设备、平台、版本和蜂窝网络。那些使用这些服务的用户只对他们用来执行测试的时间付费，也即基于运行时许可证进行付费。

2.1.3 提供定义测试自动化策略时要考虑要素的示例

许多要素可以影响到对测试自动化实现及其策略的决策。

最关键的要素是：

- 时间限制。
- 所需的开发测试自动化解决方案（TAS）的测试自动化工程师（TAE）的专业水平和人数。
- 测试硬件。
- 测试工具许可证。
- 适应性。
- 维护。
- 对不同的平台的支持（例如 Web、桌面和/或移动）。
- 对持续集成/持续交付（CI/CD）的支持。
- 对测试管理集成和测试报告功能的支持。

第一个也是最重要的成本因素是必须完成的工作和测试自动化本身的时间表。

从时间安排的角度，如果截止日期很短，通常使用的方法是只雇用几个熟练的测试自动化工程师（TAE）来创建测试自动化解决方案（TAS）。在路线图和时间表更长时，组织可以根据更多背景信息确定所需的 TAE 人数。当被测系统演变、改进并变得更加复杂时，可以决定雇用更多的 TAE 来实施和维护 TAS 以及测试。

其他重要的成本因素是工具及其许可证。预算应包含所需的测试工具、硬件和测试自动化工程师（TAE）的额外培训。这些要素与集成到其他工具和系统以执行任何额外的非测试功能（例如将测试结果上传到测试管理系统，或触发配置管理系统中的构建）有着紧密的联系。对于这些附加功能，有免费或收费的工具和库可供使用，但是在创建测试自动化策略的过程中必须考虑这些因素，并且应该包括在预算中。

一个开发团队有多少测试环境和代理（测试执行节点）总是依赖于具体周境。被测系统和发布计划越大、越复杂，组织需要的资源就越多，这也会增加成本。最近一种约束资源成本的方法是使用云提供商，按需为测试硬件资源和运行时许可证付费，但必须小心使用，因为可能会适得其反。有时，机器持续运行的成本可能比自有硬件更昂贵。这种方法可以降低高昂的维护成本和 TAE 团队的运营成本。

2.2 测试自动化中的角色和职责

2.2.1 总结成功的测试自动化方案所需的角色和技能

对于一个成功的测试自动化解决方案（TAS），组织需要熟练的测试自动化工程师（TAE），他们应该具备很强的编程和技术架构知识。

根据项目的规模和成熟度，还应该至少有一个很强的了解实际业务领域和测试目标的领域专家（Subject Matter Experts, SME）（例如测试主管、架构师和业务分析师）。此人的角色是根据测试目标和实际的测试自动化解决方案（TAS），推动并创建相关概念及路线图。为了开展实际工作，需要具备团队管理和软技能，以便培训、激励和建设团队。[ISTQB® 认证测试专家级 - 测试管理 - 管理测试团队（CTEL-TM-MTT）]。

测试自动化工程师（TAE）应具有很强的技术能力以及关于不同的 SDLC、被测系统的体系结构及其开发环境的知识。除技术方面之外，TAE 应该有能力与测试分析师和其他利益相关者就测试自动化的目标进行协作。由于穷尽测试无法实现，这个结论同样适用于测试自动化：100%的测试自动化覆盖

率也无法实现。由于时间和精力有限，TAE 需要能够从商业和投资的角度出发，通过参与风险评估来确定并优先覆盖最具影响力的测试条件。

中国软件测试认证委员会 (CSTQB®)

3. 准备测试自动化-225 分钟（K3）

关键词

API 测试（API testing），组件测试（component testing），契约测试（contract testing），左移（shift left），右移（shift right），被测系统（system under test），测试自动化方法（test automation approach），测试条件（test condition），测试替身（test double），测试级别（test level），测试金字塔（test pyramid）

第 3 章的学习目标：

3.1 跨测试级别的集成

- CT-TAS-3.1.1 (K2) 区分测试自动化的不同分布情况
- CT-TAS-3.1.2 (K2) 根据被测系统架构选择测试自动化方法
- CT-TAS-3.1.3 (K3) 展示优化测试自动化分布的方法，以实现左移和右移

3.2 不同软件开发生命周期模型中的策略考虑

- CT-TAS-3.2.1 (K2) 解释测试自动化项目如何与传统的软件开发生命周期模型相符合
- CT-TAS-3.2.2 (K2) 解释测试自动化项目如何与支持测试自动化的敏捷软件开发最佳实践相符合
- CT-TAS-3.2.3 (K3) 为测试自动化项目做好准备，使其符合支持持续测试中测试自动化的 DevOps 最佳实践

3.3 测试自动化的应用和可行性

- CT-TAS-3.3.1 (K2) 解释用于确定测试是否适合进行测试自动化的准则
- CT-TAS-3.3.2 (K2) 识别只有通过测试自动化才能解决的挑战
- CT-TAS-3.3.3 (K2) 识别难以自动化的测试条件

3.1 跨测试级别的集成

3.1.1 区分测试自动化的不同分布情况

迈克·科恩（Mike Cohn）提出了测试自动化金字塔的初始概念，该金字塔描述了三个层级：单元级（ISTQB 称之为组件级）、服务级和用户界面（UI）级。从那时起，出现了许多变种，它们都具有相同的基本原则，即描述测试级别的作用和测试用例在这些级别中的分布（参见 ISTQB CTFL 教学大纲，5.1.6 节 测试金字塔）。

在迈克·科恩（Mike Cohn）最初的测试自动化金字塔中，重点是由测试自动化解决方案（TAS）执行的测试类型。服务级可以分为三种测试类型：组件集成测试、契约测试和应用程序编程接口（API）测试。

单元测试（ISTQB 称之为组件测试）用于确认单个组件，侧重于代码质量。组件集成测试允许利用测试替身（如模拟（mock）和桩（stub））来确认用户界面和 API。契约测试可以确认服务之间的契约。API 测试侧重于通过真实服务连接使用真实数据对给定服务进行功能确认。用户界面（UI）测试是系统的端到端测试，通过图形用户界面（GUI）进行交互。

描绘测试的当前状态作为基线，并描绘目标状态，会很有帮助。这对缺少什么或可接受的测试级别是什么，提供了清晰理解。这将表明在每个测试级别上执行的测试量，以及是手动测试还是自动测试。目标状态还取决于时间表，以及在该时间框架内可实现的目标。如果可以实现，金字塔形态应该是目标形态。

测试分布示例：

- **金字塔形：**这是一种测试的平衡分布，在较高的测试级别上执行较少的测试，而在较低的测试级别上执行较多的测试，以获得稳定和更快的测试。如果可用资源和时间框架允许，那么这通常是目标状态金字塔。
- **冰淇淋甜筒形：**这是金字塔的倒置版本。服务测试的数量是均衡的，但是测试严重依赖于在用户界面（UI）测试级别中发现大多数缺陷，由于其复杂性，实现自动化的成本通常更高。由于缺乏组件测试，缺陷在软件开发生命周期（SDLC）中发现的较晚。
- **沙漏形：**测试主要集中在最高和最低的测试级别上，而服务级别测试几乎缺失，从而导致集成缺陷。如果业务逻辑由应用程序编程接口（API）（即服务）提供，那么许多用户界面（UI）测试可以很容易地移动到较低的测试级别。

- **雨伞形：**测试完全依赖于昂贵的用户界面（UI）测试。这导致缺陷修复周期变长，测试用例和测试自动化解决方案（TAS）的维护成本升高。如果在技术上无法实现较低级别的测试用例，则摆脱“伞形结构”可能无法实现，此时重点应该放在优化 UI 测试自动化套件、减少测试执行时间和提高稳定性上。

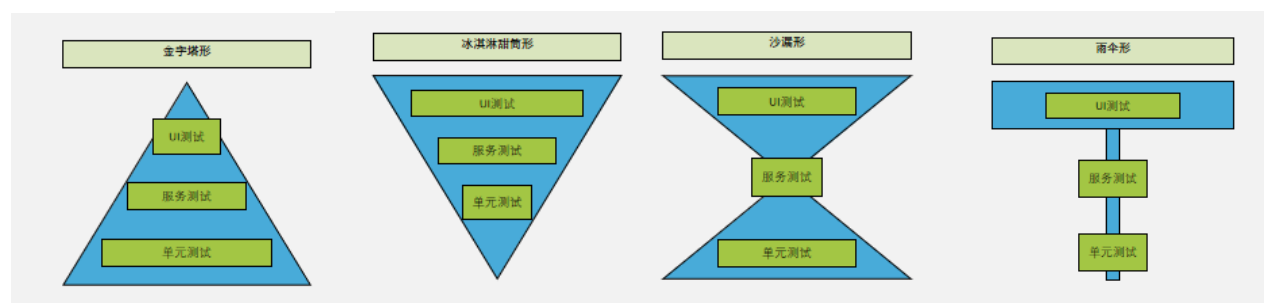


图 1：测试分布示例

3.1.2 根据被测系统架构选择测试自动化方法

在测试策略中定义测试级别可以有多种方式。选择哪一种很大程度上取决于若干要素，如组织的文化、软件工程的整体成熟度、遵循的软件开发生命周期（SDLC）以及被测系统是大型机还是微服务架构。

尽管金字塔形被认为是理想的测试分布，但并不总是可以实现的，或者需要较长的时间才能过渡到那种最终状态。将可实现的形状作为目标状态是一种很好的做法，例如从雨伞形过渡到沙漏形。一旦实现了这个目标，就可以确定新的目标状态。

为大型机系统选择适当的测试自动化分布与现代软件开发不同，并且同样取决于许多要素，因为某些级别可能根本无法实现自动化。传统上，对大型机的访问是通过终端模拟器（即终端绿屏）完成的。批处理的验证是可能的，但也是有限的。组件测试更难进行。由于几乎不存在微服务，通过应用程序编程接口（API）测试来验证接口之间的数据通信并不总是可行的。通过批处理作业、数据库和图像用户界面（GUI）进行测试更为普遍。

正在对其遗留的解决方案进行改造、慢慢转向微服务架构的组织，可以将应用程序编程接口（API）测试和契约测试引入到系统的改造部分。

3.1.3 展示优化测试自动化的分布，以实现左移和右移方法

一旦识别了当前状态分布并选择了目标状态分布，就可以策划改进路线图。这决定了用自动化测试什么（即测试自动化的范围）以及如何测试（即测试自动化策略）。需要确定一个按优先级排序的待实施项目列表和测试自动化的入口准则。

在测试自动化分布不平衡的情况下，建议采用自底向上的方法。如果代码覆盖率低，则表明缺少组件测试用例，需要编写额外的组件测试用例。第二步，需要审查组件测试的质量，并在必要时加以改进。建议的最佳实践，遵循测试驱动开发技术、举办测试技术研讨会、都可以提高自动化测试的质量。

为用户界面（UI）和应用程序编程接口（API）测试引入组件测试，利用测试替身（例如代理、桩），有助于从昂贵、低效和不可靠的测试来实现左移。消除对真实服务和数据的依赖提高了测试执行的一致性，使之易于集成到 CI/CD 流水线中提供早期反馈。

近年来，契约测试增长明显。通过验证供应者和消费者之间的契约，可以更容易更早地找到缺陷的根本原因。团队不会依赖应用程序编程接口（API）测试或用户界面（UI）测试来检测底层服务的缺陷，而是可以减少这种测试用例的数量。为 API 构建调用关系图（call graph），追踪哪些 API 相互连接以及哪些是所选 API 的消费者或生产者，是一种明智的做法。这能更好地识别系统的潜在痛点。

左移是一种在软件开发生命周期（SDLC）中将测试移到前期的做法，而右移是将测试移到后期，当被测系统发布时，右移作为在预生产测试环境或生产中评估性能的一种手段。通过使用右移，测试人员可以监控应用程序和应用程序编程接口（API）的性能，从实际用户那里获得反馈。尽管测试自动化在左移方法中更加突出，但如果与可观测性相结合，测试自动化可以帮助更快决定是否可以继续完整发布，或者候选发布是否需要回滚。

右移测试的目标：

- 了解用户偏好。
- 支持金丝雀发布和灰度发布，最大限度减少对用户功能的干扰。
- 尽早识别产品中的缺陷。
- 有助于扩展测试自动化的范围和用途。
- 提高覆盖率。

3.2 不同软件开发生命周期模型中的策略考虑

3.2.1 解释测试自动化项目如何与传统的软件开发生命周期模型相符合

在瀑布模型中，测试阶段在需求分析、系统设计和实现阶段之后。由于这个严格的顺序，测试自动化活动启动比较晚。测试之间更长的周期意味着利用测试自动化的机会更少，并且来自测试自动化的反馈通常来得太晚，导致投资回报率（ROI）更低。

在 V 模型中，所有的测试计划和文档准备都发生在早期阶段。举个例子，架构设计阶段产生了集成测试设计，比在瀑布模型中更早建立了测试。可惜地是，测试自动化解决方案（TAS）的实际实现仍然出现在软件开发生命周期（SDLC）后期，尽管与瀑布模型相比测试自动化的 ROI 更高，但仍然落后于现代敏捷软件开发实践。

3.2.2 解释测试自动化项目如何与支持测试自动化的敏捷软件开发最佳实践相符合

遵循敏捷软件开发的理想之一是实现冲刺（sprint）内的测试自动化。这意味着确定所有必要的测试自动化活动，并作为每个用户故事出口准则的一部分。包括测试用例定义、自动化测试用例的实现、更新到测试自动化框架（TAF），以及在某些情况下集成到 CI/CD 流水线。未能正确应用敏捷实践的组织，通常对测试工作量进行估算，而是要么将测试工作单独分配到一个独立的任务单中，要么完全不对其进行监控。通过实现冲刺（sprint）内的测试自动化，敏捷团队确保他们准备好在每个冲刺（sprint）内或在每个冲刺（sprint）结束时部署约定的范围。从敏捷的角度，如果团队还不成熟，那么第一个目标应该是实现冲刺（sprint）测试，而自动化会落后一个冲刺（sprint）。在此基础上，团队能以自己的方式实现冲刺（sprint）内的测试自动化。

3.2.3 为测试自动化项目做好准备，使其符合支持持续测试中测试自动化的 DevOps 最佳实践

敏捷软件开发聚焦于如何组织工作，而 DevOps 负责软件的端到端交付。这是通过自动化的构建、集成、测试、部署和生产活动来实现的。这便于通过反馈机制进行持续测试，从而确保持续改进。

重点更多地放在实现较低级别的自动化测试用例上，包括组件、组件集成和契约测试。通过减少对真实数据和服务的依赖，测试会在更短时间内执行。如果可行，测试自动化应该在被测系统构建的相同流水线中执行。在每个开发和构建阶段（例如本地、拉取请求、合并和部署）之后，会触发不同的测试套件。

如果测试人员有能力构建更大的用户界面（UI）和应用程序编程接口（API）测试套件，那么它们将被单独执行以提供额外的价值。建议将手动测试工作的重点放在探索性测试和最终用户反馈上，作为测试自动化的补充活动。

3.3 测试自动化的应用和可行性

3.3.1 解释用于确定测试是否适合进行测试自动化的准则

为测试自动化选择测试用例通常是由测试分析师完成的，他们理解哪些测试用例可以被自动化，或者由测试自动化工程师（TAE）完成，他们具备必要的领域知识来做出这样的决定。

在为测试自动化选择测试用例并确定其优先级时，要考虑以下几点：

- 以自动化的方式实现测试用例在技术上可行吗？
- 是否有任何技术挑战会影响自动化测试用例的交付？团队是否准备好并受过良好的培训来开展实现工作？
- 编码工作是否提供了足够的 ROI？（见 5.1.1 节）
- 频繁运行测试用例是否有价值？
- 是功能性测试还是非功能性测试？是冒烟测试套件、回归测试套件还是确认测试套件的一部分？
- 测试用例是否可重复？
- 当被测系统由于更新而变更时，测试用例是否易于维护？
- 测试用例是否涵盖常用的业务工作流程？
- 是否存在功能重叠的测试，从而允许测试步骤和测试数据的复用？

3.3.2 识别只有测试自动化能解决的挑战

某些测试只能通过测试自动化来执行。包括以下类别：

1. 手动测试执行时间超出合理时长。
2. 测试用例的执行需要同步。
3. 需要在流水线中使用测试结果。

4. 需要针对缺陷分析大量日志文件。
5. 测试中需要精确计时。
6. 需要跨多个操作系统、浏览器、设备、位置或配置进行测试编排。
7. 要大量的测试执行和/或数据输入来最大化覆盖率。
8. 任何需要自动监控和分析的非功能测试，或来自大量用户输入的非功能测试，如压力测试或可靠性测试。

3.3.3 识别难以自动化的测试条件

某些测试条件使得自动化测试用例成为一项困难的挑战，甚至是一项不可能完成的任务。有很多实际的例子，下面列出了其中的一些：

- 设计需求的确认，包括不同平台的用户界面（UI）一致性。软件的整体外观和感觉是主观的，需要人工审查。
- 任何涉及太多人工交互的测试用例。在金融领域，贷款申请就是一个很好的例子，在此过程中可能会出现某些规章或条件（例如，收入不足和个人详细信息不正确）。这种情况下，经纪人需要仔细检查实际贷款申请，并作出人为决定或联系客户。
- 阻塞测试自动化活动的技术难题，例如操作系统级的限制。一个例子是向用户发送文本消息的本地操作系统软件。因为操作系统不允许在目标被测系统之外进行交互，所以无法使用用户界面（UI）测试自动化工具进行这些文本消息的交互或验证。
- 具有长时间相关性的测试条件会使测试自动化效率低下，并且与手动测试执行相比，通常难以正确设置。举个例子，测试人员需要登录到被测系统，刷新主页内容并等待两个小时，以等待被测系统因会话超时而自动注销。如果不能以手动方式修改运行时间（例如桩，服务器端响应和操作系统级的时间变更），则不建议以自动化的方式实现这样的测试用例。

4. 测试自动化的组织部署与发布策略-135 分钟（K2）

关键词

组件（component），确认测试（confirmation testing），质量闸（quality gate）

第 4 章的学习目标：

4.1 测试自动化解决方案策划

- CT-TAS-4.1.1 (K2) 识别测试自动化如何支持缩短上市时间的方式
- CT-TAS-4.1.2 (K2) 识别测试自动化如何帮助根据需求验证已报告的缺陷的方式
- CT-TAS-4.1.3 (K2) 定义为运营相关的场景开发测试自动化的方法

4.2 测试自动化部署策略

- CT-TAS-4.2.1 (K2) 定义测试自动化的部署策略
- CT-TAS-4.2.2 (K2) 识别测试自动化在部署中的风险
- CT-TAS-4.2.3 (K2) 定义降低测试部署风险的方法

4.3 测试环境中的依赖关系

- CT-TAS-4.3.1 (K2) 定义测试环境中的测试自动化组件
- CT-TAS-4.3.2 (K2) 识别测试自动化的基础组件及其依赖关系
- CT-TAS-4.3.3 (K2) 为了与被测系统进行集成定义测试自动化的数据和接口需求

4.1 测试自动化解决方案策划

4.1.1 识别测试自动化如何支持缩短上市时间的方式

测试自动化有助于更快将软件推向市场，因为它有助于减少测试周期的时间。软件发布前的测试需要进行一些全面的验证和确认。这在回归测试期间尤其重要，因为这种类型的测试促进了重用，降低了成本，并保证了测试执行之间的一致性。

测试自动化有助于减少手动测试工作，在覆盖了相同的测试范围的情况下为开发人员提供了快速反馈。如果组件测试和组件集成测试是自动化的，那么它还允许在软件开发生命周期的早期进行测试，这通常不是以手工方式执行的。

质量阀是构建过程中的强制措施，软件在进入下一阶段之前需要满足该措施。基于测试自动化构建的质量阀，可以加快部署到预生产环境或生产环境的部署过程。通过利用这些质量阀，可以更早发现缺陷，从而缩短上市时间。测试执行时间可以利用并行测试执行和遵从测试左移方法来缩短。测试左移方法促进了一种质量观念的文化，鼓励在软件开发生命周期的早期进行测试。这可能包括多个独立的测试用例或跨浏览器的测试。

额外信息，请参见第 3.1.3、6.1.2 和 6.2.1 节。

4.1.2 识别测试自动化如何帮助根据需求验证已报告的缺陷的方式

在代码修复之后执行的确认测试可以处理已报告的缺陷。测试者通常按照复现缺陷必要的测试步骤来验证缺陷不再存在。

缺陷有可能在后续版本中重新出现（例如，这可能表明了配置管理或代码仓库管理的问题），因此确认测试适合进行测试自动化，并且可以添加到现有的回归测试套件中。

自动化确认测试通常只有很窄的功能范围。一旦报告了缺陷，并且理解了复现缺陷所需的测试步骤，就可以在任何时候进行实现。

跟踪自动化确认测试可以帮助报告解决缺陷所花费的时间和测试周期数。

通过跨多个平台、设备、浏览器和操作系统版本使用测试自动化来验证修复，大大减少了花费在测试上的时间。

4.1.3 定义为运营相关的场景开发测试自动化的方法

运营验收测试确保软件为用于生产系统做好了准备，通常在发布前进行。这项工作旨在对被测系统（SUT）的系统、组件以及其他基础设施进行最终确认，并测试其是否准备好投入生产。这是必要的，因为尽管测试自动化工程师（TAE）尽了最大努力，但不能保证被测系统（SUT）在测试环境之外和生产环境中会表现出相同的行为。

一个好的运营测试计划将包括对可靠性、容错性、完整性和可维护性的测试。以下是可以用于测试自动化的方法：

- 静态代码分析 — 自动化工具分析代码以查找安全漏洞。测试结果会存储起来，以便进行时间趋势分析。
- 端到端测试 — 自动化测试脚本执行端到端用户场景，并全面测试测试环境以发现任何缺陷。
- 故障转移测试 — 自动化测试脚本专门测试当应用程序的硬件离线时会发生什么。一些例子包括在物理服务器、云服务器、网络、计算机磁盘等硬件组件出现故障时应用程序如何恢复。测试脚本旨在衡量被测系统（SUT）自动恢复能力的成败，这对于实施混沌工程的组织尤为重要。
- 备份和恢复测试 — 自动化测试脚本测试当前版本备份的成功与否，然后回滚到之前的点（即软件的旧版本）。
- 性能效率测试（例如负载测试） — 自动化测试脚本可以用于模拟许多用户同时使用 SUT。存储测试结果以进行比较和随时间的趋势分析。
- 运营文档审查 — 自动化测试脚本可通过比较被测系统（SUT）文档的版本用于此活动，并根据加入到特定版本的新功能向开发团队发出是否需要更新的信号。
- 信息安全测试 — 自动化测试脚本可以与行业标准安全测试工具结合使用，以静态和动态方式评估被测系统（SUT）。测试结果会按时间存储，用于比较和趋势分析。
- 基于组织服务水平协议（SLA）的监控 — 在软件开发生命周期（SDLC）中使用的自动化测试脚本可以重新用于监控生产操作，并在自动化测试失败时发出警报。这是一种主动措施，可以在真实用户遇到生产中断之前捕捉到它们。

测试自动化对运营软件确认可以提供巨大的好处，特别是如果自动化测试可以重复执行并在多个环境中执行。可以创建专门的自动化测试套件，以便生成测试结果并与之前的测试执行进行比较。这

确保了 SUT 新版本发布时的一致性。随着时间的推移，一个可靠的运营特定测试条件的自动化测试套件可以降低成本。

4.2 测试自动化部署策略

4.2.1 定义测试自动化的部署策略

良好的测试自动化部署策略将考虑的方面包括但不限于整体测试环境、可用的测试工具、对被测对象（SUT）的访问、测试脚本存储和其他依赖项以及测试数据的供应。有了这些高层考虑，测试自动化工程师（TAE）可以开始考虑开发和部署测试自动化解决方案（TAS）的策略。

测试环境 — 测试自动化工程师（TAE）应考虑他们将如何在不同的测试环境中访问被测系统（SUT）。当他们开始开发他们的测试自动化测试软件时，无论他们使用的是关键字驱动的方法还是其他解决方案，他们必须考虑这个解决方案如何在多个测试环境中运行。例如，测试脚本应该开发成以最小的更改就能够在测试环境和预生产环境中运行。典型地，对于基于 Web 应用程序，这涉及到更改统一资源定位器（URL），然后测试脚本无论在哪个测试环境中都以相同的方式运行。

工具 — 测试自动化工程师（TAE）们还需要考虑他们用于构建测试自动化解决方案（TAS）的工具。如果它是一个商业工具，他们可能需要了解它如何授权。TAE 可能会发现他们的测试环境需要访问工具的许可服务器。当测试脚本打算在多个测试环境中使用时，需要考虑这一点。只是因为，许可服务器在测试环境中可用，并不一定意味着它在预生产环境中也会可用。

软件访问 — 需要明白的重要考虑因素是如何访问被测系统（SUT）。测试脚本可以设计为接受参数，以便当 SUT 端点更改时可以快速更新特定用户或具有特殊访问权限的凭据。这在切换到不同的测试环境、URL 需要更改时变得非常重要。此外，预生产环境可能需要使用不同的凭据（例如，通过用户和测试帐户、生物识别技术和智能卡）。一个好的自动化测试脚本设计将包括足够的灵活性，以便无论测试环境如何，都可以简单设置参数就能按预期执行测试脚本。

测试脚本存储 — 测试自动化工程师（TAE）将希望决定一个中心化的位置来存储和管理自动化测试脚本。一个好的策略将包括使用配置管理的源代码存储库。通过这种方式，只要可以访问源代码存储库，就可以从多个测试环境中访问测试脚本，并且可以为被测系统（SUT）的特定版本创建版本。所有配置和依赖项都可以保存在与测试脚本相同的存储库中，并提供最佳的可移植性。简而言之，测试自动化解决方案（TAS）、测试自动化框架（TAF）和所有测试用例都可以在存储库中存储和管理。

数据供应 — 重要的是了解自动化测试脚本是否依赖于被测系统（SUT）的测试环境中已经存在的某些测试数据。好的测试脚本设计将尽可能避免静态数据（即固定数据集）。然而，有些情况下这是

不切实际的。在这些情况下，测试自动化工程师（TAE）将需要确定一个解决方案来预加载测试数据，或者使用测试自动化创建测试脚本，以便在实际测试脚本运行和测试 SUT 之前生成所需的测试数据。两种方法都有利弊。一方面，让管理员预加载测试数据是方便的。然而，TAE 就依赖另一个资源提供能力来支持他们。通过设置测试脚本来实现自给自足，可以消除对另一个资源（管理员）的依赖。然而，这些测试脚本的收集整理需要时间，并且成为需要维护的解决方案的另一部分。

通过考虑上述所有项目，测试自动化策略将能够对测试自动化活动提供适当的规划和控制。

4.2.2 识别测试自动化在部署中的风险

技术问题可能导致产品风险和项目风险（更多信息，请参见 CTFL 教学大纲，第 5.2.2 节）。典型的技术问题包括：

- 过多的抽象可能导致难以理解测试自动化代码实际上做什么（例如，使用关键字驱动方法中的关键字）。
- 测试数据表可能变得太大/复杂/冗长，难以迁移到其他测试环境，导致不一致的状态结果。
- 依赖测试自动化解决方案（TAS）使用某些操作系统库或其他组件，这些组件可能不是在 SUT 的所有测试环境中都可用。

典型的部署项目风险包括：

- 人员问题：可能难以找到维护测试自动化的合适人员。
- 由于被测系统（SUT）更新导致测试自动化解决方案（TAS）操作不正确，需要计划外的 TAS 维护。
- 引入测试自动化的延迟。
- 根据对被测系统（SUT）的更改来更新测试自动化解决方案（TAS）导致的延迟。
- 测试自动化解决方案（TAS）无法捕获非标准用户界面（UI）对象。
- 允许过时的测试用例保留在测试套件中，浪费测试执行时间。

测试自动化解决方案（TAS）项目的潜在失效点包括：

- 迁移到不同的测试环境。
- 部署到生产环境。

- 忘记自动化是软件，也应该被测试。

重要的是要意识到各种技术问题、项目风险和潜在失效点可能危及测试自动化项目的成功。常见的技术问题包括过度抽象而产生的复杂性上升、测试数据管理的挑战以及对特定组件的依赖。项目风险还需要考虑人员配置的困难、系统更新导致的维护问题、部署延迟以及过时测试用例的存在。此外，还应通过适当的规划，解决迁移到不同环境和忽视测试自动化软件本身的需要等潜在失效点。一句话，监测和积极措施可以确保测试自动化项目的成功。

4.2.3 定义降低测试部署风险的方法

有许多风险缓解策略可以用来处理这些风险领域。以下是一些讨论。

无论是内部开发还是获取到的解决方案，测试自动化解决方案（TAS）有自己的软件生命周期（SDLC）。需要记住的一件事是，像任何其他软件一样，TAS 需要在配置管理下，并且其特性需要有文档记录。否则，部署它的不同部分并使它们协同工作或在某些测试环境中工作将变得非常困难。

此外，必须有一个有文档记录的、清晰的、易于遵循的部署规程。这个规程是依赖于版本的；因此，它也必须包含在配置管理下。

部署测试自动化解决方案（TAS）有两种不同的情况：

1. 首次部署。
2. 维护部署 - TAS 已经存在，需要部署更新。

与首次部署相关的风险包括：

- 测试套件的总测试执行时间可能比测试周期计划的测试执行时间长。在这种情况下，重要的是要确保在下一个计划的测试周期开始之前，为测试套件的完整执行留出足够的时间。
- 测试环境存在安装和配置问题（例如，数据库设置和初始加载，以及服务的启动/停止）。测试自动化解决方案（TAS）需要一个测试夹具（即预定义的数据集），以在测试环境中为自动化测试用例创建必要的前提条件。

对于维护部署，还有其他考虑因素。测试自动化解决方案（TAS）本身需要演进，它的更新需要部署到生产中。在将 TAS 的更新版本部署到生产环境之前，需要对其进行测试。因此，有必要检查新功能，验证测试套件可以在更新后的 TAS 上运行，可以发送测试报告，并且没有性能缺陷或其他质量问题。在某些情况下，可能需要更改整个测试套件以适应 TAS 的最新版本。

4.3 测试环境中的依赖关系

4.3.1 定义测试环境中的测试自动化组件

测试自动化组件通常包括工具、虚拟机、自动化测试脚本、容器和配置。测试环境还包括被测系统（SUT）。测试环境中的测试自动化组件可以定义如下：

被测系统（SUT） - 这是测试环境的一个明显部分。被测系统可以作为一个整体或者被划分为子组件进行测试（例如，API、基于 Web 的界面和数据库）。

平台 - 平台描述了测试自动化组件的托管位置。这包括可能用于使测试自动化高效和易于移植的云基础设施、网络、虚拟机和容器。

测试用例和测试套件 - 这描述了由测试步骤组成的单个测试用例，这些步骤指导自动化和手动操作。测试套件描述了测试用例的逻辑分组，以便它们可以一起高效执行。

工具 - 因各种原因使用的不同测试自动化工具。工具集合将包括用于用户界面（UI）测试自动化、测试用的应用程序编程接口（API）端点、数据生成、监控、需求管理、缺陷管理、日志记录和报告工具，以及基于度量生成趋势的工具。

测试自动化框架（TAF） - 这包括构成 TAF 设计的所有项目。TAF 包括驱动脚本、公共库、自动化测试用例模板、数据加载/生成脚本、如何使用 TAF 组件的文档，以及帮助测试自动化工程师（TAE）使用 TAF 的教程。更多信息，请参见 CTAL-TAE' 教学大纲，第 3.1 节。

维护测试组件是一个主要考虑因素，因为过度复杂化测试环境可能导致不必要的修复 TAF 缺陷的时间，而不是从解决方案中受益。非常重要的一点是：找到合适的工具、配置和平台可移植性的混合，使组件尽可能可重用。

4.3.2 识别测试自动化的基础组件及其依赖关系

在组装测试自动化时，需要意识到许多基础设施组件和依赖关系。它们共同涵盖了运行测试自动化解决方案（TAS）所需的所有先决条件。主要的组件和依赖关系包括：

主机 - 这些可以是虚拟机、物理服务器、笔记本电脑和设备（例如，平板电脑和移动设备）。它们安装了测试自动化软件，是测试脚本的创建和执行地点。

网络 - 这为测试自动化解决方案（TAS）提供了访问被测系统（SUT）的途径。它还可以包括多个主机及其网络连接，以提供自动化测试的并行执行。通常，主机需要在同一网络上，并正确配置以相互通信。

平台 - 像任何其他软件一样，测试自动化可以运行在云平台上或设计为在容器中运行。只要平台提供对底层操作系统的权限和访问，就可以安装所有必要的工具和依赖关系。

软件依赖关系 - 为了使测试自动化工具正确运行，有必要了解测试自动化的所有其他依赖关系。例如，特定的测试自动化工具可能需要先在主机上安装最新版本的编程语言。在选择工具之前和之后，测试自动化工程师（TAE）需要考虑这些依赖关系。

SUT - 一旦考虑并正确配置了组件、依赖关系和全部基础设施，最后一步是确保对被测系统（SUT）的访问。除了网络之外，还需要考虑如何与被测系统交互。例如，在基于 Web 的应用程序中，需要在主机上安装浏览器。需要考虑浏览器的类型和版本。

4.3.3 为了与被测系统进行集成定义测试自动化的数据和接口需求

测试自动化工程师（TAE）在开发测试自动化脚本之前需要考虑的两个问题是，他们计划如何与 SUT 交互以及测试用例的数据依赖性是什么。以下是一些例子：

应用程序编程接口（API） - 如果被测对象（SUT）使用了 API，可以在端点级别进行测试，而不需要在应用层面通过用户界面（UI）进行交互。这可能需要更深入地理解被 UI 隐藏的端点和数据通信。测试自动化工程师（TAE）需要理解以什么顺序调用 API 来创建业务流程，以及如何关联数据以保持测试用例的完整性。暴露在互联网上的 API 也被称为 Web API 或 Web 服务。

数据库接口 - 一些测试自动化工具能够与 SUT 底层数据库直接交互。可以编写测试脚本来验证列和行中的数据，确保存储过程和其他数据库规则配置正确。这可能需要数据库中已经存在特定的数据值用于可靠性测试。

接口兼容性 - 使用契约测试确保两个独立的系统（例如，两个微服务）是兼容的，并且能够相互通信。契约测试可以是消费者驱动的、生产者驱动的，也可以是双向的。更多细节可以在国际软件测试认证委员会高级测试自动化工程师（CTAL-TAE）教学大纲的第 5.1.3 节找到。

此外，测试自动化工程师（TAE）必须仔细考虑他们将如何与被测系统（SUT）交互，并理解测试用例中的数据依赖性。无论是通过应用程序编程接口（API）、数据库接口进行测试，还是确保系统间的接口兼容性，关注这些考虑因素对于健壮和有效的测试自动化至关重要。通过深思熟虑地解决这些因素，TAE 可以提高他们自动化工作可靠性和效率，最终有助于 SUT 的质量和成功。

5. 测试自动化影响分析-150 分钟（K3）

关键字

覆盖率（coverage），测试报告（test report）

第五章学习目标：

5.1 建立和维护测试自动化的投资

CT-TAS-5.1.1 （K3） 展示构建测试自动化解决方案的投资回报。

5.2 测试自动化度量

CT-TAS-5.2.1 （K2） 对测试自动化度量进行分类。

5.3 测试自动化在项目和组织层面的价值

CT-TAS-5.3.1 （K3） 确定使用测试自动化的组织级考虑因素。

CT-TAS-5.3.2 （K3） 分析有助于确定测试自动化测试目标的最佳实施方式的项目特征。

5.4 根据测试自动化报告做出决策

CT-TAS-5.4.1 （K2） 分析测试报告数据引导决策制定。

5.1 建立和维护测试自动化的投资

5.1.1 展示构建测试自动化解决方案的投资回报

在开始项目实施之前，估算测试自动化的建立和维护成本非常重要。除了自动化可以为项目带来的价值外，了解项目的投资回报（ROI）计算也是有益的。

投资回报率（ROI）计算可以在项目生命周期的任何给定时刻提供有意义的反馈，通过展示某项活动的相对于投入的工作量所带来的回报。ROI 计算可以进一步调整为简单地将花费的时间乘以各自的劳动成本，包括手工测试人员成本和测试自动化工程师（TAE）成本。

要计算 ROI，需要确定测试自动化的投资（即时间和成本）以及通过它实现的节省：

$$\text{ROI} = \text{节省} / \text{投资}$$

需要注意的是，节省的成本和投资可以根据不同的度量和数据，以不同的测量方式和单位来计算。在此教学大纲范围内，使用了一个简单的模型来演示此方法，以时间单位而不是成本进行衡量。如果某些活动仅以成本衡量，那么可以使用特定于项目的比率将其转换为时间。

通常，测试自动化得到的节省是因为相同的测试可以比手动运行时间显著缩短。这也意味着它们可以更频繁地运行。因此，可以增加执行的测试的次数。

要计算节省，需要考虑以下度量指标：

- 手动运行测试用例的时间。
- 运行自动化测试用例的时间。
- 测试用例的数量。
- 测试运行的次数。

要计算投资，需要考虑以下度量指标：

- 建立测试自动化的时间。
- 开发自动化测试脚本的平均时间。
- 实现的自动化测试脚本数量。
- 自动化测试脚本的平均维护时间。
- 运行自动化测试脚本的时间。

- 自动化测试脚本失败的百分比。
- 定义的测试用例数量。
- 测试运行次数。

通过为敏捷软件开发调整上述模型，可以如下图所示预测项目的冲刺。可以使用该图表来确定测试自动化从哪个冲刺(Sprint)开始收回了其投资。

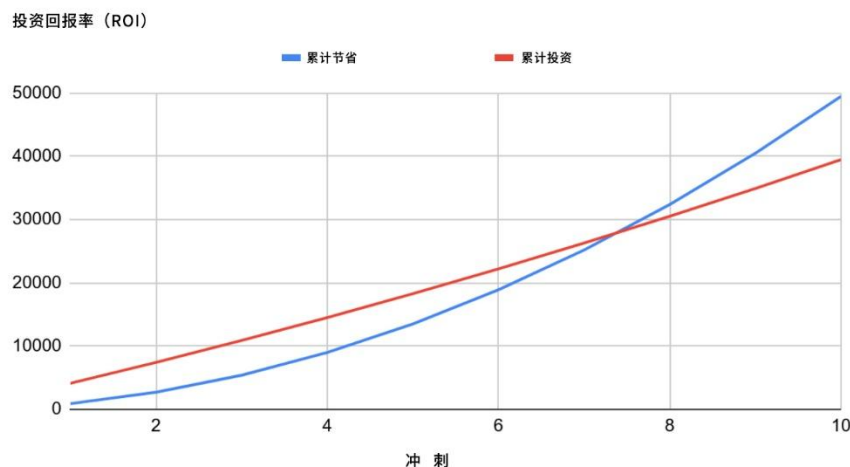


图 2：投资回报率（ROI）计算示例，显示投资回报点

请记住，在计算 ROI 时测量的度量指标之间的某些联系，包括：

- 如果计划的项目持续时间少于 ROI 的转折点，引入测试自动化是不值得的。在这种情况下，通过手动执行测试，可以节省时间和投入。
- 由于投资在很大程度上取决于自动化测试的执行时间，应用测试金字塔并在正确的测试级别实现测试用例可以减少执行时间并提高 ROI。

5.2 测试自动化度量

5.2.1 对测试自动化度量进行分类

基于事实数据的趋势分析有助于决策的制定。通过收集测试自动化解决方案（TAS）的度量指标，测试人员能够评估 TAS 并对以下事项做出决策：

- TAS 对项目的适用性。

- TAS 对新测试条件扩展功能的适应性。
 - 被测系统（SUT）中的用户流程变化。
 - 测试执行方式的变化。
- 由于在 TAS 中发现的缺陷而导致的测试自动化维护性。

测量的成本应尽可能低，这通常可以通过自动化收集和报告度量指标来实现。以下是一些例子：

通过/失败比率

这是一个常见度量指标，跟踪通过的自动化测试与未达到预期结果而失败的自动化测试之间的比例。通过/失败比率 = 通过的测试数量 / 失败的测试数量。

失效/缺陷的比率

自动化测试的一个常见问题是，许多测试可能因为同一个原因而失败，例如 SUT 中的单个缺陷。测量因特定缺陷而失败的自动化测试数量可以帮助指出可能存在的问题。此外，可以收集每个缺陷的失败次数，并跟踪失败的原因：子系统缺陷、整个系统缺陷、测试数据问题或基础设施问题。

测试自动化执行时间

一个更容易确定的度量指标是执行自动化测试所需的时间。在测试自动化解决方案（TAS）的初期，这可能不重要，但随着自动化测试用例数量的增加，这个度量指标可能变得非常重要。这个度量指标包括 TAS 的构建时间。

自动化测试用例的数量

这个度量指标可以用来展示测试自动化项目所取得的进展。但必须考虑到自动化测试用例的数量并没有透露太多信息：例如，它并不指示覆盖率水平。

测试自动化的功能覆盖率

这个覆盖率指示了自动化测试用例覆盖的功能需求的百分比。

代码覆盖率

代码覆盖率跟踪低级别（即组件）测试执行的代码行数。

没有绝对的百分比表明足够的覆盖率，而且在除了最简单的软件之外，通常很难达到 100%的代码覆盖率。然而，普遍认为覆盖率越高越好，因为这增加了对被测系统（SUT）的信心。随着增加了更多的低级别自动化测试，预计代码覆盖率将增加。

5.3 测试自动化在项目和组织层面的价值

5.3.1 确定使用测试自动化的组织级考虑因素

在任何项目开始测试自动化之前，需要在组织内识别以下因素：

软件开发的策略和实践

检查开发团队的工作方式和现有的文档类型是很好的做法。任何可用的文档都有助于确定如何将测试自动化接入到开发团队的流程中。文档可以包括被测系统（SUT）的技术规格说明、使用的软件 and 开发工具，或任何关于开发的策略，如代码审查指南、定义的编码标准和合并流程。

现有活跃的测试自动化项目及其状态

在进行中的开发项目中，可能有不同团队正在进行的一个或多个测试自动化解决方案（TAS）开发项目。对于决策者来说，一般建议是检查这些现有项目及其状态，看看它们中的任何一个是否符合新 TAS 的既定测试目标。通过分析解决方案，可以确定是重用现有的 TAS 还是根据当前需求创建一个新的 TAS。

组织中测试自动化的领域专家

建议寻找能够协助在组织内推广新测试自动化解决方案（TAS）的领域专家（SME）。SME 可以分享他们关于 TAS 及其推广的故事和经验教训。从中，可以识别出为成功推广 TAS 需要考虑或避免的不同风险。

测试环境的可用性

对于大型组织来说，一般建议是收集关于现有测试环境、它们的使用情况和可用性的信息。这些信息至关重要，可以避免在没有通知或达成一致的情况下推出新的测试自动化解决方案（TAS）并使用该系统，从而破坏其他团队的工作。通常，项目的需求将需要一个新的测试环境，因此现有的环境可以作为创建新环境的基线，并且如果负责的团队和人员能够确定并且有空，相关工作可以更容易进行。

测试工具和许可证

了解组织目前拥有哪些工具和许可证总是值得的。通过识别这些工具和许可证，可以减少 TAS 的规划、成本和时间。而且，新项目可能已经有可用的工具。例如，如果已经通过指定的提供商设置了云测试，并且新项目有可用的许可证，那么使用不同的云提供商就没有意义。建议使用相同的工具和许可证以降低项目成本。

5.3.2 分析有助于确定测试自动化之测试目标的最佳实施的项目特征

存在一些主要的项目特征，它们可以定义一种最优的工作方式，并有助于确定成功的测试自动化目标。

领域

理解每个领域在法规或标准上的差异非常重要。例如，与金融或医疗保健领域相比，旅游领域有不同的法规和风险。始终建议检查不同的标准和领域限制，以确保计划中的测试自动化目标符合规定。

平台

在测试自动化之测试目标方面，也强调评估项目覆盖的平台以及在何处进行测试自动化是有益的。规划多个平台的测试自动化可能更加困难，因为可能需要多个解决方案来覆盖不同的平台。在许多情况下，例如对于移动应用和 Web，可以使用相同的工具。但规划将决定测试自动化解决方案（TAS）的可重用性。

编程语言和技术栈

项目的实现细节，如其编程语言和技术栈，也决定了组织应该定义什么样的测试自动化测试目标。建议与特定项目上开发人员正在使用的相同的编程语言。这样做可以使协作更容易，通过一起进行代码审查的交叉学习将进一步提高被测系统（SUT）的质量和测试自动化工程师（TAE）的知识。

项目的成熟度

通过分析项目的成熟度，可以得出设计理想测试自动化目标的信息。通过识别不同的因素，如测试用例的数量、现有的 CI/CD 流水线、测试人员和测试自动化工程师（TAE）的数量，可以构建出不同的测试目标和路线图。例如，如果有很多优先级高的手动测试用例，并且它们的手动测试执行时间很长，那么首先实现这些测试的自动化以节省时间和精力是非常重要的。除了上述因素，项目的时间线也非常重要。如果项目很短或将很快结束，没有太多时间来实施，那么规划一个大型和健壮的测试自动化解决方案（TAS）是不值得的。然而，如果项目是一个新领域项目，那么制定一个基于最小可行产品的逐步增量路线图是值得的。

利益相关者的认同

很多时候，由于关键利益相关者不认同，测试自动化没有得到利用。这可能是由于担心速度低于目标、紧迫的截止日期、预算或其他问题。在分析项目成熟度后，如果有空间开始实施测试自动化解决方案（TAS），战略利益相关者需要识别产品风险，列出引入测试自动化的好处，并提出适当的计划。这个计划需要突出通过测试自动化实现的里程碑，并指出如何提高被测系统（SUT）的可测试性，以便成功引入一个能解决已识别风险的 TAS。最后，这个计划需要呈现给利益相关者。

团队知识和相关经验

成功的测试自动化最重要和相关的要素是测试人员的技巧和经验。与测试人员合作，并利用他们的知识来定义他们和业务分析师都感到舒适的测试目标是有益的。通常，测试经理或测试组长会创建一个能力和技能矩阵，以识别团队内可用的知识，并同时识别差距。这些差距可以通过不同的目标（如培训）来识别，并分配实施任务。

测试管理支持和预算

依赖于项目的规模和成熟度，在定义测试自动化的测试目标时应该考虑可用的预算和测试管理支持。定义可以实现的测试目标是重要的，这反过来将能够获得测试管理的支持。

在向管理层提出测试自动化策略建议时，文档需要简洁，清晰地定义当前已识别的差距或未来适当地实施测试自动化的成本。一份列出建议及其收益的清单，指出业务价值和成本节省，将有助于测试管理批准开发或改进测试自动化解决方案（TAS）。

质量特性

在 ISTQB CTFL 教学大纲第 2.2.2 节“测试类型”中列出了 ISO/IEC 25010:2011 定义的质量特性。这些质量特性随后可以用于评估当前的测试自动化解决方案（TAS），或确定 TAS 将收集的度量指标。

5.4 根据测试自动化报告做出决策

5.4.1 分析测试报告数据引导决策制定

测试自动化报告的格式和内容可能因接收报告的利益相关者而异。它可以为管理层、运营或技术利益相关者创建。更多的信息可以在国际软件测试认证委员会高级测试自动化工程师（CTAL-TAE）教学大纲的 6.1.3 节找到。

收到测试自动化报告后，可以将其纳入更广泛的测试报告中，或者将其整合后在组织结构内逐级上报。

不同的利益相关者在此类测试自动化报告中找到不同的价值。一种策略方法是通过强调这些重要的度量指标，来识别那些对参与的利益相关者重要的关键度量指标。

通过自动化收集的数据可以帮助：

- 识别趋势和执行根本原因分析。
- 将测试自动化工作转移到维护。

- 将测试自动化工作转移到测试自动化框架（TAF）的改进和进一步开发。
- 增加整体测试自动化解决方案（TAS）的能力。
- 增加测试的左移和右移方法。
- 在未来的冲刺中扩大测试自动化的功能覆盖范围。
- 更多关注缺陷集群。
- 通知开发人员需改进代码的区域。
- 就整体软件开发生命周期（SDLC）流程提供建议。
- 改变后续测试自动化报告的内容和格式。

基于上述信息，测试自动化工程师（TAE）与其他利益相关者合作，可以识别现有测试自动化覆盖范围和测试结果中的差距和某些改进点。

6. 测试自动化的实现和改进策略 – 150 分钟 (K3)

关键字

覆盖率（coverage），前置条件（precondition），测试套件（test suite）

第六章学习目标：

6.1 从手动测试到持续测试的迁移活动

CT-TAS-6.1.1 (K2) 描述从手动测试迁移到测试自动化的要素及策划活动。

CT-TAS-6.1.2 (K2) 描述从测试自动化迁移到持续测试的要素及策划活动。

6.2 整个组织的测试自动化策略

CT-TAS-6.2.1 (K2) 进行测试自动化资产和实践的评估，以确定改进领域。

6.1 从手动测试到持续测试的迁移活动

6.1.1 描述从手动测试迁移到测试自动化的要素及策划活动

从手动测试迁移到测试自动化的最简单的机会是以回归测试为目标。随着当前的功能性和非功能性测试成为未来的回归测试，回归测试套件将会增长。回归测试的数量超过传统手动测试团队可用的时间和资源只是时间问题。

迁移成本

在从手动测试迁移到自动化测试期间，因为手动测试和自动化测试同时进行，可以预见项目会增加成本。一旦自动化测试被认为足以替代或取代手动测试执行活动，可以将更多的工作转移到探索性测试和为测试自动化定义额外的测试用例，这与手动回归测试的成本不同。

功能重叠

在测试脚本开发人员在不同的测试用例中包含相同的测试自动化步骤时，就会发生功能重叠。例如，大多数测试用例将以登录序列的测试步骤开始。这可能包括输入用户名、密码和选择登录按钮。在每个测试用例中添加它会增加维护活动。如果登录过程中增加了额外的测试步骤，就必须在每个测试用例中更新相同的变更。更好的方法是将登录过程制作成可重复的测试自动化组件，并让所有测试用例引用该功能。有关流模型模式的详细信息，请参见国际软件测试认证委员会高级测试自动化工程师（CTAL-TAE）教学大纲的 3.1.5 节。

数据共享

测试经常共享测试数据。当测试使用相同的测试数据记录来执行不同的 SUT 功能时，就会发生这种情况。一个例子是测试用例“A”，它验证员工的可用假期时间，而测试用例“B”验证员工作为职业发展目标所参加的课程。每个测试用例都使用相同的员工，但验证不同的参数。在手动测试环境中，员工测试数据通常会在每个手动测试用例中复制多次，这些测试用例使用此员工来验证员工测试数据。然而，在自动化测试中，应该在可能和可行的情况下存储和从单一来源访问共享的测试数据，以避免重复或引入错误。

测试相互依赖性

在执行复杂的回归测试时，一个测试可能依赖于一个或多个其他测试。这种情况可能非常普遍。例如，作为测试步骤的结果，会创建一个新的“订单 ID”。后续测试可能想要验证：a) 新订单是否正确显示在系统中，b) 是否可以更改订单，或 c) 删除订单是否成功。在每种情况下，第一个测试中动态

创建的“订单 ID”值必须被捕获以供后续测试重用。这个问题的解决取决于测试自动化解决方案（TAS）的设计。如果后续测试用例无法找到“订单 ID”，则它们将失败。

测试执行前提条件

测试自动化工程师（TAE）经常在未首先了解确保测试用例可以可靠执行的前提条件时就开始测试脚本的开发。当尝试在多个测试环境中对同一被测对象（SUT）执行测试用例时，这可能变得非常具有挑战性。使测试用例可重复的前提条件的例子包括用户名、账户角色、数据表值和其他特定数据条目。一个好的策略将包括前期分析，以了解在自动化特定测试用例之前需要存在于 SUT 中的信息。此外，通过在实际测试用例运行之前自动化创建前提条件，可以增强策略以节省时间。这可能包括运行前提条件测试脚本，这些脚本从用户界面（UI）填充 SUT 或自动化批量处理过程将测试数据加载到数据库。

功能覆盖率

识别测试中的功能缺口，这些缺口可以成为测试自动化的候选对象，如第 3 章所解释的。将 100% 的手动测试用例自动化并不代表所有可以使用测试工具进行自动化的测试用例也达到了 100%。随着越来越多的测试自动化，测试人员可以减少测试执行时间，他们可以用这些时间来识别额外的被测系统（SUT）测试以增加覆盖率。

可执行的测试

在将手动回归测试转换为自动化回归测试之前，重要的是要验证手动回归测试是否正确运行。这为确保成功转换为自动化回归测试提供了正确的起点。如果手动回归测试没有正确执行，可能是因为它编写得不好、使用了无效的测试数据、过时或与当前被测系统（SUT）不同步，或者是因为 SUT 缺陷。在理解和/或解决失败的根本原因之前就自动化它，将创建一个不工作的自动化回归测试，这是浪费和徒劳的。重要的是要展示新的自动化测试带来的等效功能，以传递对将替换旧的手动测试的自动化测试的信心。

6.1.2 描述从测试自动化迁移到持续测试的要素及策划活动

持续测试涉及比传统软件开发生命周期（SDLC）中更频繁地使用测试资产。做法是，总是在代码更改完成，代码在测试环境中可以使用后，立即启动运行测试套件。这有助于提供即时反馈，并通过在过程中尽早发现缺陷来降低缺陷成本。通过复杂的开发工具，有意愿在构建过程中尽早测试，能够做到这一点。

为持续测试进行调整测试自动化解决方案（TAS）需要遵循：

- 需要更改测试套件以运行在更新的 TAS 上时：在将它们部署到 TAS 之前，对测试套件进行必要的更改并测试它们。
- 测试中使用的桩、驱动程序和接口需要更改以适应更新的测试自动化解决方案（TAS）时：在将它们部署到 TAS 之前，对测试用具进行必要的更改并测试它们。
- 基础设施需要更改以适应更新的测试自动化解决方案（TAS）时：评估需要更改的基础设施组件。
- 更新的测试自动化解决方案（TAS）有额外的缺陷或性能缺陷时：对风险与收益进行分析。如果发现的缺陷使得更新 TAS 变得不可能，最好不要继续更新或等待 TAS 的下一个版本。如果与纠正它们的好处相比，缺陷是可以忽略不计的，TAS 仍然可以更新。
- 确保创建已知缺陷的发布说明，通知 TAE 和其他利益相关者，并尽量估计何时可以修复这些缺陷。

本节中提到的所有要点在使用 CI/CD 时变得尤为重要。将流水线整合在一起并自动化构建过程，与正在开发的测试自动化解决方案（TAS）自然适配。如果构建编排工具在正确的测试环境中，流水线可以扩展成包括自动化测试，以在部署后立即验证被测系统（SUT）。这要求测试自动化工具配置正确，可以访问 SUT，并且它可以访问代码库和访问数据准备脚本。

6.2 整个组织的测试自动化策略

6.2.1 进行测试自动化资产和实践的评估，以确定改进领域

就像任何其他开发活动一样，有策略地暂停测试自动化解决方案（TAS）开发，并寻找重构解决方案的机会是重要的。要监控的领域包括初始实施、维护以及从可重复性角度评估解决方案的能力。一些值得跟踪的良好类别包括：开发测试自动化解决方案（TAS）所花费的小时数、修复 TAS 所花费的小时数，以及与手动测试相比节省的测试人员执行时间。如果 TAS 维护时间大于手动测试时间，就表明出了问题。

在开始首次部署测试自动化解决方案（TAS）之前，重要的是要确保它可以在自己的环境中运行，它与随机更改隔离，并且可以更新和管理测试用例。TAS 及其基础设施都必须得到维护。在首次部署的情况下，需要以下基本步骤：

- 代码覆盖工具指示组件测试套件覆盖了多少代码，以及哪些地方存在可以通过额外的组件测试来填补的缺口。

- 通过创建需求可追溯性矩阵，可以确定被测系统（SUT）的功能覆盖，揭示哪些功能尚未被任何测试用例覆盖。
- 定义测试自动化解决方案（TAS）在项目或整个组织内一致的基础设施使用方式。
- 为测试套件开发一致的配置管理策略。
- 利用国际软件测试认证委员会高级测试自动化工程师（CTAL-TAE）教学大纲 3.1.4 节中描述的最佳实践，创建通用的测试自动化解决方案（TAS）开发指南。
- 实现前提条件。通常，在设置前提条件之前无法执行测试。这些前提条件可能包括选择正确的数据库或测试数据以进行测试，或设置初始值或参数。许多建立测试前提条件所必需的初始化步骤可以自动化。如果在执行测试之前这些步骤不可能遗漏，这将获得更可靠的和可信赖的解决方案。随着回归测试转换为自动化测试，这些前提条件需要成为测试自动化过程的一部分。

当进行新功能或维护目的的增量测试自动化解决方案（TAS）更新时，应考虑以下要素：

- 评估为测试自动化解决方案（TAS）提供扩展功能的测试工具的更新或其他新的测试工具。
- 评估进一步优化测试自动化解决方案（TAS）功能和性能的方法。
- 确定进一步分解和模块化测试脚本以增强可重用性的机会。
- 保证对可重用组件的认识和一致应用。
- 收集潜在改进领域的证据，并提供建议及其好处。
- 评估并纠正功能重叠领域。在自动化现有回归测试时，识别测试用例之间存在的任何功能重叠，并在可能的情况下重用先前开发的测试自动化组件。
- 评估将额外的手动测试用例进行自动化的机会，并为实现创建待办事项。
- 强调测试设计和测试数据管理改进机会。
- 确保所有现有的测试套件都适配到测试自动化解决方案（TAS）的最新版本。
- 如果测试自动化导致流水线排队，请减少集成测试的范围，只包括最关键的测试，例如创建一个冒烟测试套件。较大的回归测试套件可以单独触发或按需执行。

7. 参考文献

标准

测试自动化的标准包括但不限于：

来自 IEEE（电气和电子工程师协会）的自动化测试标记语言（ATML）包括：

- IEEE Std 1671.1:测试描述
- IEEE Std 1671.2:测试工具描述
- IEEE Std 1671.3:被测单元描述
- IEEE Std 1671.4: 测试配置描述
- IEEE Std 1671.5:测试适配器描述
- IEEE Std 1671.6: 测试站描述
- IEEE Std 1641:信号和测试定义
- IEEE Std 1636.1:测试结果

ISO/IEC 30130:2016 (E)软件工程 - 软件测试工具的能力

ETSI（欧洲电信标准协会）和 ITU（国际电信联盟）的测试和测试控制符号（TTCN-3）包括：

- ES 201 873-1: TTCN-3 核心语言
- ES 201 873-2: TTCN-3 表格表示格式（TFT）
- ES 201 873-3: TTCN-3 图形表示格式（GFT）
- ES 201 873-4: TTCN-3 操作语义
- ES 201 873-5: TTCN-3 运行时接口（TRI）
- ES 201 873-6: TTCN-3 控制接口（TCI）
- ES 201 873-7: 结合 TTCN-3 使用 ASN.1
- ES 201 873-8: 结合 TTCN-3 使用 IDL
- ES 201 873-9: 结合 TTCN-3 使用 XML
- ES 201 873-10: TTCN-3 文档
- ES 202 781: 扩展：配置与部署支持
- ES 202 782: 扩展：TTCN-3 性能和实时测试
- ES 202 784: 扩展：高级参数化
- ES 202 785: 扩展：行为类型
- ES 202 786: 扩展：支持具有连续信号的接口
- ES 202 789: 扩展：扩展的 TRI 接口

OMG（对象管理组）的 UML 测试配置文件（UTP），规定了以下测试规范概念：

- 测试架构
- 测试数据
- 测试行为
- 测试记录
- 测试管理

ISTQB®文件

标识	引用
ISTQB-AL-TM	ISTQB 认证测试工程师，高级大纲，测试经理，版本 2.0，2012 年 10 月，可通过[ISTQB-Web]获得
ISTQB-EL-TM-MTT	ISTQB 认证测试工程师，专家级/测试管理/测试团队管理，版本 2.0，2011 年 11 月，可通过[ISTQB-Web]获得
ISTQB-FL	ISTQB 认证测试工程师，基础级大纲，版本 4.0，2023 年 4 月，可通过[ISTQB-Web]获得
ISTQB-PT	ISTQB 认证测试工程师，性能测试大纲，2018 年 12 月，可通过[ISTQB-Web]获得
ISTQB-TAE	ISTQB 认证测试工程师，测试自动化工程师大纲，2024 年 2 月，可通过[ISTQB-Web]获得
ISTQB 术语	ISTQB 术语表，可通过[ISTQB-Web]在线访问

书籍

Paul Baker, Zhen Ru Dai, Jens Grabowski and Ina Schieferdecker, “Model-Driven Testing: Using the UML Testing Profile”, Springer 2008 edition, ISBN-10: 3540725628, ISBN-13: 978-3540725626
Efriede Dustin, Thom Garrett, Bernie Gauf, “Implementing Automated Software Testing: how to save time and lower costs while raising quality”, Addison-Wesley, 2009, ISBN 0-321-58051-6
Efriede Dustin, Jeff Rashka, John Paul, “Automated Software Testing: introduction, management, and performance”, Addison-Wesley, 1999, ISBN-10: 0201432870, ISBN-13: 9780201432879
Mark Fewster, Dorothy Graham, “Experiences of Test Automation: Case Studies of Software Test Automation”, Addison-Wesley, 2012
Mark Fewster, Dorothy Graham, “Software Test Automation: Effective use of test execution tools”, ACM Press Books, 1999, ISBN-10: 0201331403, ISBN-13: 9780201331400
Boby Jose, “Test Automation, a manager’s guide”, September 2021, ISBN: 9781780175478
James D. McCaffrey, “.NET Test Automation Recipes: A Problem-Solution Approach”, Apress, 2006 ISBN-13:978-1-59059-663-3, ISBN-10:1-59059-663-3
Daniel J. Mosley, Bruce A. Posey, “Just Enough Software Test Automation”, Prentice Hall, 2002, ISBN-10: 0130084689, ISBN-13: 9780130084682
Casey Rosenthal, “Chaos Engineering: System Resiliency in Practice by Casey Rosenthal”, April 2020, ISBN-13: 1492043869
Colin Willcock, Thomas Deiß, Stephan Tobies and Stefan Keil, “An Introduction to TTCN-3” Wiley, 2nd edition 2011, ISBN-10: 0470663065, ISBN-13: 978-0470663066

文章

Robert V. Binder, Suzanne Miller, “Five Keys to Effective Agile Test Automation for Government Programs” August 24, 2017, Software Engineering Institute, Carnegie Mellon University, https://resources.sei.cmu.edu/asset_files/Webinar/2017_018_101_503516.pdf
DoD CIO, Modern Software Practices “DevSecOps Fundamentals Guidebook: Activities & Tools”, Version 2.2, May 2023, https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsActivitesToolsGuidebookTables.pdf?ver=_SylglWJB9K0Jxb2XTvzDQ%3d%3d
Naveen Jayachandran, “Understanding roi metrics for software test automation”, 2005, https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=3937&context=etd
Thomas Pestak, William Rowell, PhD, “Automated Software Testing Practices and Pitfalls”, September 2018, https://www.afil.edu/stat/statcoe_files/Automated%20Software%20Testing%20Practices%20and%20Pitfalls%20Rev%201.pdf
Andrew Pollner, Jim Simpson, Jim Wisnowski, “Automated Software Testing Implementation Guide for Managers and Practitioners”, October 2018, https://www.afil.edu/stat/statcoe_files/0214simp%20%20AST%20IG%20for%20Managers%20and%20Practitioners.pdf

8. 附录 A - 学习目标/知识认知级别

下面定义的学习目标适用于本大纲。大纲中的每个主题都将根据其学习目标进行考试。学习目标根据与其对应的知识认知级别的行为动词开头，如下所示。

等级 1: 理解 Understand (K2) - 考生要能选择与主题相关陈述的原因或解释，并能对测试概念进行总结 (summarize)、比较 (compare)、分类 (classify) 和举例 (give examples)。

行为动词: 分类、比较、区分、辨别、解释、举例、演绎、总结

示例:

- “按其目的和支持的测试活动对测试工具进行分类。”
- “比较不同测试级别。（可以用来寻找相似性、差异性或二者兼而有之。）”
- “区分测试与调试。（寻找概念之间的差异。）”
- “辨别项目风险和产品风险。（允许两个（或更多）概念分别分类。）”
- “解释周境对测试过程的影响。”
- “举例说明为什么需要进行测试。”
- “从给定的失效简介中推断缺陷的根本原因。”
- “总结工作产品评审过程的活动。”

等级 3: 应用 Apply (K3) - 考生能在遇到一个熟悉的任务时执行规程，或者选择正确的规程并应用于给定的环境中。

行为动词: 应用、实施、准备、使用

示例:

- “从给定的需求应用边界值分析导出测试用例。（应参考一个规程/技术/过程等）”
- “实施度量收集方法以支持技术和管理需求”
- “准备测试自动化环境”
- “使用可追溯性来监控测试进度，以确保测试目标、测试策略和测试计划的完整性和一致性。
（在学习目标（L0）中，可用于要求考生能够使用某种技术或规程。类似于“应用”。）”

参考文献（针对学习目标的认知水平）：

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

中国软件测试认证委员会 (CSTQB®)

9. 附录 B 业务成果与学习目标的追溯矩阵

本节列出了测试自动化策略专家业务成果与测试自动化策略专家学习目标之间的追溯性。

业务成果测试自动化策略专家			B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12	B13	B14	B15
Chapter 1	测试自动化策略的简介和目标																
1.1	测试自动化项目的成功因素																
1.1.1	解释测试自动化的目标和相关性	2	X														
1.1.2	识别测试自动化项目成功的技术因素	2	X														
1.1.3	总结选择测试自动化候选项目进行投资的适用准则	2	X														
Chapter 2	测试自动化资源																
2.1	实现测试自动化解决方案的成本和风险																
2.1.1	在所有权成本方面比较替代技术方案	2		X													
2.1.2	解释测试自动化工具软件许可模式的考虑事项	2		X													
2.1.3	提供定义测试自动化策略时要考虑要素的示例	2		X													
2.2	测试自动化中的角色和职责																
2.2.1	总结成功的测试自动化方案所需的角色和技能	2			X												
Chapter 3	准备测试自动化																
3.1	跨测试级别的集成																

业务成果测试自动化策略专家			B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12	B13	B14	B15
3.1.1	区分测试自动化的不同分布情况	2				X											
3.1.2	根据被测系统架构选择测试自动化方法	3				X											
3.1.3	展示优化测试自动化分布的方法，以实现左移和右移	2				X											
3.2	不同软件开发生命周期模型中的策略考虑																
3.2.1	解释测试自动化项目如何与传统的软件开发生命周期模型相符合	2					X										
3.2.2	解释测试自动化项目如何与支持测试自动化的敏捷软件开发最佳实践相符合	2					X										
3.2.3	为测试自动化项目做好准备，使其符合支持持续测试中测试自动化的 DevOps 最佳实践	3					X										
3.3	测试自动化的应用和可行性																
3.3.1	解释用于确定测试是否适合进行测试自动化的准则	2						X									
3.3.2	识别只有通过测试自动化才能解决的挑战	2						X									
3.3.3	识别难以自动化的测试条件	2						X									
Chapter 4	测试自动化的组织部署与发布策略																
4.1	测试自动化解决方案策划																
4.1.1	识别测试自动化如何支持缩短上市时间的方式	2							X								
4.1.2	识别测试自动化如何帮助根据需求验证已报告的缺陷的方式	2							X								
4.1.3	定义为运营相关的场景开发测试自动化的方法	2							X								
4.2	测试自动化部署策略																
4.2.1	定义测试自动化的部署策略	2								X							

业务成果测试自动化策略专家			B 0 1	B 0 2	B 0 3	B 0 4	B 0 5	B 0 6	B 0 7	B 0 8	B 0 9	B 1 0	B 1 1	B 1 2	B 1 3	B 1 4	B 1 5
4.2.2	识别测试自动化在部署中的风险	2								X							
4.2.3	定义降低测试部署风险的方法	2								X							
4.3	测试环境中的依赖关系																
4.3.1	定义测试环境中的测试自动化组件	2									X						
4.3.2	识别测试自动化的基础组件及其依赖关系	2									X						
4.3.3	为了与被测系统进行集成定义测试自动化的数据和接口需求	2									X						
Chapter 5	测试自动化影响分析																
5.1	建立和维护测试自动化的投资																
5.1.1	展示构建测试自动化解决方案的投资回报	3										X					
5.2	测试自动化度量																
5.2.1	对测试自动化度量进行分类	2											X				
5.3	测试自动化在项目和组织层面的价值																
5.3.1	确定使用测试自动化的组织级考虑因素	3												X			
5.3.2	分析有助于确定测试自动化之测试目标的最佳实施方式的项目特征	3												X			
5.4	根据测试自动化报告做出决策																
5.4.1	分析测试报告数据引导决策制定	2													X		
Chapter 6	测试自动化的实现和改进策略																

业务成果测试自动化策略专家			B 0 1	B 0 2	B 0 3	B 0 4	B 0 5	B 0 6	B 0 7	B 0 8	B 0 9	B 1 0	B 1 1	B 1 2	B 1 3	B 1 4	B 1 5
6.1	从手动测试到持续测试的迁移活动																
6.1.1	描述从手动测试迁移到测试自动化的要素及策划活动	2														X	
6.1.2	描述从测试自动化迁移到持续测试的要素及策划活动	2														X	
6.2	整个组织的测试自动化策略																
6.2.1	进行测试自动化资产和实践的评估，以确定改进领域	3															X

10 . 附录 C - 版本说明

ISTQB®测试自动化策略教学大纲 2024 是一个新的 ISTQB 教学大纲，它结合了 2016 年发布的先前 ISTQB 测试自动化工程师教学大纲中的策略内容，并加入了实现测试自动化的当前最佳实践以及衡量其成功与否的额外内容。因此，没有按章节和部分给出详细的版本说明。

中国软件测试认证委员会 (CS100®)

11 . 附录 D - 领域专用术语

术语名称	定义
金丝雀发布	一种部署和测试策略，旨在通过仅向少数用户发布软件来降低风险并验证新软件。
容器	打包了代码及其依赖项的软件单元，因此软件可以在各种环境中快速可靠地运行。
DevOps	一种将软件开发和信息技术运维的工作整合并自动化的实践方法，旨在改进并缩短软件开发生命周期。
流模型模式	一个工作域、其中的组件以及它们之间连接关系的高级视图。

12. 索引

All terms are defined in the ISTQB® Glossary (<http://glossary.istqb.org/>).

所有的术语都定义在 ISTQB® 的词汇表中 (<http://glossary.istqb.org/>).

应用程序编程接口测试, 22, 23, 24

测试级别, 22, 23, 44

金丝雀发布, 75

测试金字塔, 22, 44

组件, 22, 23, 24, 25, 26, 29, 30, 46, 54, 58

测试报告, 41, 42, 51

组件测试, 22, 23, 25

测试套件, 27, 30, 32, 36, 37, 53, 54, 58, 60

确认测试, 29

容器, 75

契约测试, 22, 23, 24, 25, 40

覆盖, 16, 21, 24, 25, 27, 41, 46, 52, 53, 56, 58

DevOps, 13, 22, 26, 70, 75

流模型模式, 54, 75

前置条件, 53, 55, 56

质量阀, 29, 30

左移, 13, 22, 25, 30, 52

右移, 13, 22, 25, 52

被测系统, 13, 22, 29

测试自动化工程师 TAE, 8, 9, 16, 17, 19, 21, 26, 31, 32, 33, 34, 38, 40, 42, 51, 54, 59, 62

测试自动化框架 TAF, 17, 26, 34, 38, 52

测试自动化解决方案 TAS, 3, 8, 10, 15, 17, 18, 19, 20, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 45, 46, 47, 48, 49, 50, 51, 52, 53, 55, 57, 58, 59, 60

测试自动化方法, 22

测试自动化解决方案, 10, 17, 18, 41

测试条件, 22

测试替身, 22