文档编号：SY0001_ISTQB_CT_AuT_EN2018_CN2.2

认证测试工程师
基础级专业领域汽车软件测试工程师
大纲

版本：EN2018_CN2.2

发布日期：2023年7月10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

版权申明

英文版权声明

版权所有®2017，已注册的德国测试委员会（GTB）。

作者和德国测试委员会均已同意以下使用条款：

- 在确认和声明版权所有者和参考来源的情况下，任何个人和培训机构都可以使用本课程大纲作为培训教材。此外，只有在获得ISTQB®委员会成员的认证后，才能将本课程大纲用于市场宣传。
- 在声明参考来源和版权归本大纲的作者和德国测试委员会所有的情况下，任何个人或机构都可以将本课程大纲作为编写文章、书籍或其他出版物的基础。
- 本大纲中的所有内容均受版权保护。除非《德国版权法》（UrhG）明确允许，否则只有在获得授权人员的批准后才能使用本大纲。对于复制、改写、翻译、缩印、在电子系统中保存和处理，以及公开发布本大纲中内容的情况，需严格遵守这项版权声明。

中文版权申明

未经许可，不得复制或抄录本文档内容。

版权标志©国际软件测试认证委员会中国分会（以下简称“CSTQB®”）

注册商标

- CTFL®是已注册的德国测试委员会（GTB）在欧盟地区的注册商标。
- GTB®是已注册的德国测试委员会（GTB）在欧盟地区的注册商标。
- ISTQB®是国际软件测试资质认证委员会的注册商标。
- Automotive SPICE®是德国汽车工业协会（VDA）的注册商标。

免责声明与责任限制

关于本文档中所含信息在技术上的准确性或是否严格遵守任何适用法律、政府法规或规定，ISTQB®不作任何陈述或担保。此外，对于适销性或适用于任何特定目的或不侵犯知识产权的情况，不作任何陈述或保证。对于因使用本文档造成的任何利益损失、意外事故或间接损害，无论在任何情况下，ISTQB®或GTB均不承担任何责任。ISTQB®和GTB明确表示，任何使用或依赖本文档提供的信息所造成的风险均由用户自行承担。本文档未推荐或暗示任何产品或供应商。

中国软件测试认证委员会（CSTQB®）

修订历史

版本	日期	备注
1.0	2011/1/19	作者：Hendrik Dettmering博士应gasq GmbH 的请求编写本课程大纲,版权已完全转让给德国测试委员会。
1.1	2015/6/14	审核内容，并与德国ISTQB®基础级认证测试工程师课程大纲 2011年版本1.0.1和ISTQB®术语表版本2.2进行比较，按照 GTB工作组于2015年3月15日慕尼黑会议的指示进行发布。
2.0	2017/3/31	在版本1.1的基础上对学习目标和内容进行了修订。 按照GTB工作组于2017年3月31日在美茵河畔的法兰克福召开会议的指示发布德文版本课程大纲。
2.0.1（英文版）	2017/6/30	只对关键术语进行了细微修订；在2017年3月第1次内部初审后，根据国际审核人员提出的意见（主要是文字方面）进行修订（请参阅“致谢”部分），插入了相应的英语参考文献（请参阅“参考文献”部分）。
2.0.1（英文版）	2017/8/13	在对ISTQB® WG术语表进行讨论后，对术语进行了细微修改；在2017年7月第2次内部初审后，根据国际审核人员提出的意见（主要是文字方面）进行修订（请参阅“致谢”部分）。
2.0.1（英文版）	2017/8/20	再次对ISTQB® WG术语表进行讨论后，对术语进行了细微修改；改正了审核人员最新发现的问题。
2.0.1（英文版）	2017/9/15	改正了GTB工作组慕尼黑会议期间审核人员发现的问题。
2.0.1（英文版）	2017/9/16	对第3.2.2章进行了修订。
2.0.1（英文版）	2017/9/22	对GA Beta测试版草案进行最终编辑。
2.0.2（英文版）	2018/5/28	根据Beta测试版审核结果进行最终编辑，可用于GA发布。
2.0.2（英文版）	2018/7/4	在GA批准后去除了水印，并添加了商标使用限制，可用于 ISTQB®出版。
EN2018_CN1.0	2020/8/28	英文版大纲2.0.2版本地化完成。
EN2018_CN2.0	2023/1/12	中文大纲修改和完善。
EN2018_CN2.1	2023/3/23	针对大纲中提及的“考试的形式为多选题”更正为“考试的形式为单选题”。
EN2018_CN2.2	2023/7/10	对个别错误进行修改。

中国软件测试认证委员会（CSTQB®）

目录

中国软件测试认证委员会（CSTQB®）

中国软件测试认证委员会（CSTQB®）

致谢

德国测试委员会（GTB）对2017年德语版课程大纲（版本2.0）的作者和审核团队表示诚挚的感谢，具体名单如下（按字母顺序排列）：

Graham Bath、André Baumann、Arne Becher、Ralf Bongard（课程大纲负责人兼工作组联合主席）、Kai Borgeest、Tim Burdach、Mirko Conrad、KlaudiaDussa-Zieger、Matthias Friedrich、Dirk Gebrath、Thorsten Geiselhart、Matthias Hamburg、UweHehn、Olaf Janßen、Jacques Kamga、Horst Pohlmann（考试负责人兼工作组主席）、Ralf Reißing、Karsten Richter、Ina Schieferdecker、Alexander Schulz、Stefan Stefan、Stephanie Ulrich、JorkWarnecke和Stephan Weißleder。

德国测试委员会（GTB）和汽车软件测试工程师工作组要对2018年英文版课程大纲（版本2.0.x）的扩展审核团队表示感谢，团队成员包括：Graham Bath、Thomas Borchsenius、Ádám Bíró、ZsoltCsatári、Attila Farkas、Attila Fekete、FerencHamori、Ádám Jezsoviczki、GáborKapros、Miguel Mancilla、Roland Milos、Kenji Onishii、MiroslawPanek、MirosławPanek、BarthomiejPredki、Stefan Stefan、Stuart Reid、Ralf Reissing、Hidetoshi Suhara、Tamás Széplakin、EshrakaZakaria和Csaba Zelei。

本课程大纲中文2020版翻译参与者（按姓氏拼音排序）：

崔哲、黄颖华、李晶（组长）、李云轩、眭佳彩、王聪、邢悦

本课程大纲中文2020版 QA 评审参与者（按姓氏拼音排序）：

胡尧、刘伟、卢志坚、陆俊文

本课程大纲中文2023版修改和完善参与者（按姓氏拼音排序）：

周震漪

致谢企业：博彦集智科技有限公司

中国软件测试认证委员会（CSTQB®）

本课程大纲的历史

1.0版课程大纲是Hendrik Dettmering博士应全球软件质量协会AISBL（gasq）的请求在2010年/2011年编写的。

为了对本文档进行审核，我们任命了来自OEM的几位专家，他们在对本课程大纲的质量和目标进行审核和评估后，认为本课程大纲符合要求。因此，将本文档作为汽车软件测试工程师认证的课程大纲，同时也作为认证的培训材料和编写考试试题的基础。

从2014年1月1日起，德国测试委员会（GTB）的“汽车软件认证测试工程师”工作组负责本课程大纲的进一步编写工作，以便尽快确定主题并满足行业要求，不仅建立行业独立的核心课程大纲，还可以作为ISTQB®基础级汽车测试工程师关于汽车软件测试领域的参考资料。

版本1.1是在2015年6月15日发布的。此版本与1.0版兼容；在1.1版中删除了与ISTQB®基础级课程大纲冗余部分。

中国软件测试认证委员会（CSTQB®）

大纲简介[1]

本文档的目的

本课程大纲对国际软件测试资质认证委员会（以下简称ISTQB®）之软件测试培训计划中的基础级汽车测试工程师进行了定义。培训机构根据该课程大纲编写课程教材，并确定合适的教学方法。学员也根据本课程大纲准备参加认证考试。

有关本课程大纲的历史和背景的更多信息，请参阅本课程大纲的历史部分。

ISTQB® CTFL®-专业（模块）：汽车软件测试工程师

目前的基础级专业模块认证汽车测试工程师培训模块面向参与汽车领域软件测试的所有人员，包括：测试员、测试分析员、测试工程师、测试顾问、测试经理、软件发布测试工程师和软件开发人员。基础级相关人员还包括希望了解汽车领域软件测试基础知识，希望对这一领域有初步了解的人员，例如，项目经理、质量经理、软件开发经理、系统分析员（业务分析员）、IT经理或管理顾问等。

中国软件测试认证委员会（CSTQB®）

[1]文本的主要部分摘选自ISTQB® CTFL®核心课程大纲[21]。

业务价值

在这部分，我们将简单介绍具备CTFL®汽车软件测试工程师认证的测试工程师能够带来的业务价值。

作为一名认证的CTFL®汽车软件测试工程师（CTFL®-AuT）应具备以下能力：

AUTFL - BO - 01	在测试团队中能进行高效协作。（“协作”）
AUTFL - BO - 02	能根据特定的项目要求，有效使用和适配ISTQB®基础级认证测试工程师（CTFL®）已掌握的测试技术。（“适配”）
AUTFL - BO - 03	能根据相关标准（Automotive SPICE®、ISO 26262等）的基本要求，选择合适的测试技术。（“选择”）
AUTFL - BO - 04	能支持测试团队以风险为导向规划测试活动，并应用已有的结构化和优先级划分知识。（“支持和应用”）
AUTFL - BO - 05	能在测试环境中应用虚拟测试方法（例如 HiL-硬件在环系统、SiL-软件在环系统、MiL-模型在环系统等）。（“应用”）

学习目标/认知水平

本课程大纲中的每段内容都指定了认知水平要求：

- K1：记住。
- K2：理解。
- K3：应用。
- K4：分析。

学习目标明确了学员在完成相应段落/章节/模块的学习后应掌握的内容。

培训机构可在时间允许的情况下讲授学习目标中标记为[扩展知识]的内容，但这些内容不在考试范围内。

示例： AUTFL - 2.2.3.1回顾ISO 26262的设计和结构。[扩展知识]

术语

在标题“术语”（K1）下的段落中列出的所有术语，即使在学习目标中没有明确提到这些内容，学员也应该能够记住并复述，这也适用于ISTQB®术语表中的定义和经过批准的版本中对术语的翻译（包括本课程大纲中的附加术语）。

中国软件测试认证委员会（CSTQB®）

考试

根据本课程大纲内容有附加的针对专业领域的认证考试，即基础级专业汽车软件测试工程师认证考试。一道考试题可以涉及课程大纲的多个章节。通常，每一考题都与一个学习目标相对应，但是那些与关键术语相关的考题除外。考试的形式为单选题。考试可以在完成认证培训课程后直接进行，也可以单独进行考试（例如，在考试中心或公开的考试）。参加培训课程不作为参加考试的先决条件。

参加考试的要求

要参加汽车软件测试工程师认证考试，考生必须拥有ISTQB®基础级认证测试工程师（CTFL®）证书，并且对汽车软件开发项目的测试感兴趣。

但是，建议考生具备以下条件：

- 至少有软件开发或软件测试方面的基础背景知识（例如，具有六个月的系统测试工程师或验收测试工程师或开发人员经验）。
- 或参加过经 ISTQB®标准认证（由 ISTQB®委员会成员）的培训课程和/或
- 已经具备汽车行业 E/E 开发项目测试的基本经验。

授权

如果培训机构使用的课程教材遵循本课程大纲，则ISTQB®委员会成员可以对该培训机构进行授权。培训机构应从提供授权服务的委员会或机构那里获取授权指南。经过认证的课程即被认为遵守本课程大纲，可以包含做为单独部分的附加考试。

培训机构可在附录中找到更多参考。

详细程度

详细程度可确保培训课程和考试内容完全一致。为了达到这个目标，本课程大纲包含以下几方面内容：

- 一般学习目标，描述（扩展）基本等级的目的。
- 必须学习的内容，包含描述以及其他进一步的文献（根据需要）。
- 每个知识领域的学习目标，描述学员在经过培训后要取得的、对知识的认知结果和思维模式。
- 学员应该能够理解和掌握的术语。
- 对要学习的重要概念的描述，包括已被广泛认可的技术文献、规范和标准等来源。

中国软件测试认证委员会（CSTQB®）

本课程大纲并不是对“汽车电子开发项目中面向软件系统的测试”知识领域进行全面讲解，只是提供与学习目标相关的必要知识范围和详细程度。

课程大纲结构

本课程大纲主要由四个章节组成。章节标题指明本章最具挑战性的学习目标/最高认知水平，同时还规定了本章在认证课程中的最短培训时间。

示例：

简介（K2）[30分钟]

该示例表明，对于“简介（K2）”一章，要求达到K1[3]和K2级别（而不是K3级别），讲授本章培训内容的计划时间为30分钟。

每章都包含多个子章节，每个子章节也会规定相应的学习目标和教学时间。如果没有给出子章节的教学时间，表明教学时间已包含在主章节中了。

中性用语

为了方便阅读，我们避免使用区分性别的语言，例如男性用户和女性用户。为了平等起见，所有角色名称均适用于两种性别。

中国软件测试认证委员会（CSTQB®）

[3] 高级别的学习目标要求了对低级别学习目标的要求。

1. 简介（K2）[30分钟]

关键词

无测试专业术语

学习目标

AUTFL-1.1.1	（K2）	解释并举例说明在汽车产品开发中，由于项目目标的多样化和产品复杂性的日益增长而带来的挑战。
AUTFL-1.2.1	（K1）	回顾项目中受标准影响的方面，如时间、成本、质量和项目/产品风险。
AUTFL-1.3.1	（K1）	根据ISO/IEC 24748-1[1]回顾系统生命周期中的六个常规阶段。
AUTFL-1.4.1	（K1）	回顾测试工程师能够在发布过程中参与的协作和贡献。

简介

软件测试的七大原则之一是“测试活动依赖于测试的内容”[21]。本章内容简单介绍了“汽车软件测试工程师”[4]所处的E/E开发环境。一方面，多样化的目标、日益增加的复杂性，以及巨大的创新压力给测试工作带来了巨大的挑战。另一方面，测试工程师在各类标准和汽车生命周期形成的框架内工作。总之，测试工程师在种种挑战和限制下，为软件和系统的发布做出了重要贡献。

中国软件测试认证委员会（CSTQB®）

[4] 在下文中，我们将只使用术语“测试工程师”，是“汽车E/E软件测试工程师”的简称。

1.1 项目目标的多样性和日益增长的产品复杂性带来的需求（K2）[15分钟]

汽车制造商和供应商比过去更快地推出新的车型[5]，而同时面临日益增加的成本压力。以下几方面会对汽车开发过程产生影响：

- 车型种类和复杂性不断增加：

  为了更好地满足终端客户的需求，OEM（汽车制造商）推出了越来越多的车型。然而，这使得每种车型的数量却在减少。为了弥补由此增加的开发和生产成本，制造商常常基于同一平台开发出多种车型。由于同一平台需要兼容多款车型，因此，其开发过程要比开发单一车型复杂得多。

- 功能范围不断扩大：

  终端客户要求汽车具备越来越多的创新功能，但又不能减少现有功能，从而导致功能范围不断扩大。

- 配置数量不断增加：

  终端客户希望能够根据个人意愿调整自己的个性化汽车。这就要求一款车型必须具备多种可能的配置，以及不同的功能。

- 提高了质量要求：

  尽管增加了功能要求和复杂性，但终端客户仍希望汽车及其功能的质量能够保证不变甚至更高。

由于项目的目标时间、成本和质量是相互冲突的（“项目管理三角形”/“魔法三角形”），因此，汽车制造商（OEM）和供应商必须努力寻找一种更加高效的系统开发方法，能随着不断增加的复杂性、不断提升的质量要求，以及在较少预算的情况下去缩短开发时间。

1.2 项目哪些方面会受到标准的影响（K1）[5分钟]

标准会对项目的几个主要方面产生影响，如时间、成本、质量、项目风险和产品风险：

- 标准可通过以下方式提高过程的效率（例如，在保持稳定质量的同时减少开发时间和/或成本）：
  - 统一命名。
  - 提高透明度。

[5] 示例来自管理咨询公司Progenium进行的一项调查研究，内容为：“1990年，只提供101种不同的车型，而到了2014年，这个数字已增加到453种”[43]。

中国软件测试认证委员会（CSTQB®）

  - 便于协作（内部和外部）。
  - 增加可重用性。
  - 经验的积累（“最佳实践”）。
- 凭借成熟和完善的技术指南[21]，这些标准有助于尽早识别和消除风险和缺陷。
- 标准为审计提供了依据，评审员可以根据标准评估产品或过程的质量。同时，评审员还可以检查产品或过程是否符合要求[1]。
- 标准是合同或监管条款和准则的一部分。

在本课程大纲中将探讨以下规范和标准：

- 有专门对过程和方法进行标准化的标准，如ISO 26262[3]或Automotive SPICE（ASPICE）[2]。
- 有专门对产品进行标准化的标准，如AUTOSAR[3]。

1.3 系统生命周期中的六个常规阶段（K1）[5分钟]

一辆汽车以及汽车的所有组件（部件）的系统生命周期[6]都是从产品概念开始，以退役而终止。在整个生命周期过程中，涉及到开发过程、业务过程、物流过程，以及与制造技术有关的工艺过程。根据预先定义的入口和出口准则，明确的里程碑有助于实现成熟的过程。这里将系统生命周期[7]分为六个阶段[1]并使之同步，在括号中的为典型测试活动[8]：

- 概念（测试计划）。
- 开发（测试分析、设计、实施、执行、评估和报告）。
- 生产（终检/下线测试）。
- 使用（无测试活动）。
- 维护（维护测试）。
- 退役（迁移测试）。

汽车行业一般产品开发过程简单概括为：概念、开发和生产。

[6] 电控单元（硬件和软件）以及组件。
[7] ISO 26262的安全生命周期包含类似的阶段。
[8] ISO 26262的安全生命周期包含类似的阶段。

中国软件测试认证委员会（CSTQB®）

1.4 测试工程师在发布过程中的贡献/参与（K1）[5分钟]

在汽车开发过程中，通过声明一个正式发布来表明项目达到了一个里程碑，并在随后看到了相关证据，由此决定了目标已经实现。从此刻开始，发布的子系统就满足了其使用的成熟度要求和目标。

发布过程用来指导各发布项的发布。发布项应包括已测试项（例如参数化的软件配置，必要时还包括硬件和机械部分）以及其他支持文档。

测试工程师通过最终测试报告提供有关发布过程的重要信息[3]：

- 已测试项和性能特性，包括其版本。
- 已知缺陷。
- 产品度量。
- 当达到测试出口准则时，按照发布规则（例如最佳实践指南引出的）提供发布建议（即：在封闭道路或公共道路上进行测试、安装建议）。

此外，测试工程师还要参与创建与之后发布相关的交付内容[4]：

- 参与变更的决策并确定其优先级。
- 划分功能的优先级（实现的顺序）。

中国软件测试认证委员会（CSTQB®）

2. E/E系统测试标准（K3）[300分钟]

关键词

Automotive SPICE（ASPICE）

Automotive SPICE（ASPICE），软件合格性测试（software qualification test）（ASPICE），系统合格性测试（system qualification test）（ASPICE）。

ISO 26262

汽车安全完整性等级（Automotive Safety Integrity Level）（ASIL），功能安全（functional safety），方法列表（method table）（ISO 26262）。

AUTOSAR

无测试专业术语

比较

无测试专业术语

学习目标

Automotive SPICE（ASPICE）

AUTFL-2.1.1.1	（K1）	回顾Automotive SPICE（ASPICE）的两个维度。
AUTFL-2.1.1.2	（K1）	回顾ASPICE的3个过程类别和8个过程组[扩展知识]。
AUTFL-2.1.1.3	（K2）	阐述ASPICE的能力级别0，1，2，3。
AUTFL-2.1.2.1	（K1）	回顾ASPICE的5个测试相关过程的目的。
AUTFL-2.1.2.2	（K2）	从测试的角度阐述ASPICE的四个评级和能力指标的含义。
AUTFL-2.1.2.3	（K2）	阐述ASPICE对测试策略的要求，包括回归测试策略。
AUTFL-2.1.2.4	（K1）	回顾ASPICE对测试文档的要求。
AUTFL-2.1.2.5	（K3）	设计单元验证的验证策略（区别于测式策略）和准则。
AUTFL-2.1.2.6	（K2）	从测试角度阐述ASPICE的各种可追溯性要求。

中国软件测试认证委员会（CSTQB®）

ISO 26262

AUTFL-2.2.1.1	（K2）	阐述E/E系统功能安全的目标。
AUTFL-2.2.1.2	（K1）	回顾测试工程师对安全文化的贡献。
AUTFL-2.2.2.1	（K2）	根据ISO 26262介绍测试工程师在安全生命周期中所发挥的作用。
AUTFL-2.2.3.1		回顾ISO 26262的结构和设计[扩展知识][9]。
AUTFL-2.2.3.2	（K1）	回顾ISO 26262中与测试工程师有关的卷名称（对应部分的标题）。
AUTFL-2.2.4.1	（K1）	回顾ASIL重要等级。
AUTFL-2.2.4.2	（K2）	阐述ASIL对适用测试技术和测试类型的影响，包括静态和动态测试，以及测试的范围。
AUTFL-2.2.5	（K3）	能够阐述ISO 26262的方法列表。

AUTOSAR

AUTFL-2.3.1	（K1）	回顾AUTOSAR的目标。
AUTFL-2.3.2	（K1）	回顾AUTOSAR的总体设计[扩展知识][10]。
AUTFL-2.3.3	（K1）	回顾AUTOSAR对测试工程师工作的影响。

比较

AUTFL-2.4.1	（K1）	回顾ASPICE目标和ISO 26262目标的不同之处。
AUTFL-2.4.2	（K2）	阐述ASPICE、ISO 26262和CTFL® 中关于测试级别的不同之处。

中国软件测试认证委员会（CSTQB®）

[9] 不在考试范围之内。
[10] 不在考试范围之内。

2.1 Automotive SPICE（ASPICE）（K3）[140分钟]

简介

过程改进遵循的原则是系统的质量取决于开发过程的质量。在这种情况下，通过与参考模型的比较来衡量组织的过程能力，过程模型提供了改进过程的方法。此外，根据评估结果，过程模型还可以用作改进组织过程的框架[5]。

2001年起，SPICE[11]用户组和AUTOSIG（Automotive Special Interest Group，汽车特殊利益集团）共同编制了Automotive SPICE（ASPICE）。自2005年发布以来，该标准已在汽车行业站稳脚跟。

2015年7月，德国汽车工业协会（VDA）发布了3.0版ASPICE[9]。从2017年起，ASPICE的改进版V.3.1将替代[6]现有的2.5版本[2]。因此，本段中的描述均参考ASPICE 3.1版[47]。

2.1.1 ASPCIE的设计和结构（K2）[25分钟]

2.1.1.1 ASPICE的两个维度

ASPICE从两个维度定义了评估模型：

在**过程维度**，ASPICE定义了过程参考模型。这些模型可以当作参考，用来比对组织过程，以便能够对组织过程进行评估和改进。对于每个过程，ASPICE 都定义了其目的和结果，以及所需的行动（基本实践）和工作结果（工作产品）。如果组织需要ASPICE以外的其他参考过程，可以从标准ISO/IEC 12207[10]或ISO/IEC 15288[11]中获取。

在**能力维度**，ASPICE定义了许多过程属性。这些属性介绍了过程能力的可测量特征。对于每个过程，都有特定于过程的属性和通用属性。ISO/IEC 33020可作为评估过程能力的依据[39]。

通过这个模型，可以对过程（过程维度）的能力（能力维度）进行评估。

2.1.1.2 过程分类（从过程维度）

ASPICE将过程分为8个过程组，这些组又分成3个过程类[9][47]：

主要过程是与公司核心过程相关的过程：

- 产品和/或服务的采购/获取（ACQ-Acquisition）。
- 产品和/或服务的供应（SPL-Supply）。
- 系统工程（SYS-System engineering）。
- 软件工程（SWE-Software engineering）。

[11] “软件过程改进和能力测定”的缩写。

中国软件测试认证委员会（CSTQB®）

支持过程是为其他过程提供支持的过程：

- 支持过程（SUP-Supporting）。

组织过程是为公司目标提供支持的过程：

- 项目或过程的管理（MAN-Management）。
- 过程改进（PIM-Process improvement）。
- 系统和组件的复用（REU-Reuse）。

对于测试工程师来说，系统开发（SYS）和软件开发（SWE）这两个过程组尤其重要。这两个过程组构成了Automotive SPICE V模型的过程（[9]附录D“重要概念”）。

2.1.1.3 能力维度中的能力级别

评估师通过一个包含六个级别的评估系统对过程能力进行评估（以级别显示）。ASPICE对能力级别0级到3级[12]的定义如下[9][47]：

- L0（过程不完整）：过程不存在或不能实现过程的目标。示例：测试工程师只检查了小部分的需求。
- L1（已执行过程）：所执行的过程实现了过程目标（但执行方式与原计划可能不一致）。示例：没有测试过程的完整计划。但是，测试工程师可以展示需求的满足程度。
- L2（已管理过程）：项目对过程进行了规划，并在执行过程中进行了监督。在某些情况下，为了实现目标，会在执行过程中调整行动计划。确定了对工作产品的要求。项目成员检查了工作产品并进行了核准。示例：测试经理确定了测试目标，规划了测试活动，并监督了测试过程。如有偏差，他会采取相应的措施。
- L3（已确定过程）：项目采用标准化的过程，并使用结果和反馈不断改进和提升。示例：测试经理为整个组织制定了一个通用测试策略。测试完成后（请参阅基础测试过程），测试经理会进一步对它进行开发和改进。

中国软件测试认证委员会（CSTQB®）

[12] 能力级别4和5目前不在汽车行业重点关注范围之内。

2.1.2 标准的要求（K3）[115分钟]

2.1.2.1 测试特定过程

ASPICE根据软件和系统开发的所有过程对测试过程进行了定义[8]：

- 软件单元验证（SWE.4）过程需要进行静态和动态测试。此过程会根据其详细的设计（SWE.3）对软件的组件进行评估。
- 软件集成测试（SWE.5）会根据软件架构设计对集成的软件进行评估（SWE.2）。
- 软件合格性测试（SWE.6）会根据软件需求对集成的软件进行评估（SWE.1）。
- 系统集成测试（SYS.4）会根据系统架构设计对集成的系统进行评估（SYS.3）。
- 系统合格性测试（SYS.5）会根据系统需求对集成的系统进行评估（SYS.2）。

2.1.2.2 评估等级和能力指标

评估师可以通过能力指标来评估过程能力。ASPICE为能力等级（过程）定义了9个过程属性（PA）。对于能力级别1到3，其定义如下（括号中以SWE.6为例）[9]，[47]：

- PA1.1：过程执行（测试工程师通过基本测试过程确定自己的工作）。
- PA2.1：执行管理（测试工程师负责计划、监督和控制包括测试活动在内的一些相关活动）。
- PA2.2：工作产品管理（测试工程师负责检查包括测试文档在内的一些文档的质量）。
- PA3.1：过程定义（负责测试过程的人员定义通用和跨项目的测试策略）。
- PA3.2：过程部署（测试工程师应用PA3.1中定义的测试策略）。

对于过程执行（PA1.1），ASPICE定义了两类指标：基本实践（BP-Base Practices）和工作产品（WP-Work Products）。此外，还定义了通用实践（GP-Generic Practices）和资源。过程属性的评估基于这些指标的实现程度，设定为四个评估等级[9]，[47]：

- N（无/None）：　　未实现（0%至≤15%）。
- P（部分/Partly）：　　部分实现（>15%，≤50%）。
- L（大部分/Largely）：　　大部分实现（>50%，≤85%）。
- F（完全/Fully）：　　完全实现（>85%，≤100%）。

要使过程达到特定能力等级，该能力等级对应级别的指标必须是“大部分实现（L）”，而低于对应级别的能力等级指标则必须是“完全实现（F）”。

中国软件测试认证委员会（CSTQB®）

2.1.2.3 测试策略和回归测试策略

作为基本实践，ASPICE要求为每个测试专用过程制定测试策略[13]（请参阅2.1.2.1）。测试经理负责在测试计划中制定测试策略。测试指南、项目目标以及合同和法规要求是制定测试策略时要考虑的基本要素。

测试工程师应知晓尽早测试的测试原则。这项原则也适用于汽车环境中的软件测试。然而，这里采用这个原则还有另一个原因，那就是较高测试级别的测试环境成本显著增加。例如，要进行较高级别的测试，专门开发完成的嵌入式硬件是必须的（例如，产品原型或唯一的模型）。测试策略不仅要定义与测试级别相适应的测试环境，还要定义测试工程师需要在哪些测试环境中执行哪些测试。

回归测试策略是测试策略的一个重要组成部分。其挑战在于如何选择经济适用的测试用例（“测试的附加价值”）。回归策略定义了选择回归测试的目标和技术。例如，可以基于风险进行选择。影响分析可帮助确定测试工程师在回归测试中需要关注的方面。但是，测试经理也可能会要求测试工程师对每次发布都重复执行所有的自动化测试用例。

2.1.2.4 ASPICE中规定的测试文档

关于测试活动的文档，ASPICE要求测试工程师编写一些工作产品（WP），正如CTFL®规定的一样[9]：

- WP08-50：测试规格说明（包含测试设计、测试用例和测试规程说明）。
- WP08-52：符合ISO/IEC/IEEE 29119-3[34]要求并包含测试策略（WP19-00）的测试计划。
- WP13-50：测试结果、测试日志、事件/偏差报告和测试总结报告。

对于每项工作产品，ASPICE都定义了其特性和内容示例。评估师可以通过现场抽查来评估这些工作产品。对于评估师来说，这些工作产品是评估过程执行情况的客观指标。

关于测试计划，ASPICE直接引用了ISO/IEC/IEEE 29119-3[14]标准。该标准还提供了可用于其他所需工作产品的模板，并且可以根据特定目的对这些模板进行调整。必须确保测试计划的内容能够帮助实现过程的既定目标。

[13] 根据 CTFL® [2]，项目专用测试策略也称为测试技术。
[14] 这取代了ISTQB® 课程大纲中仍在使用的IEEE 829:1998和IEEE 829:2008。

中国软件测试认证委员会（CSTQB®）

2.1.2.5 单元验证的验证策略和准则（SWE.4）

对于软件单元的验证（SWE.4），ASPICE 要求制定验证策略[15]。在SWE.5/SWE.6/SYS.4/SYS.5这些与测试相关的过程中，ASPICE要求制定测试策略（请参阅2.1.2.3）。这个测试策略“只”针对动态测试。因此，它是对代码评审和静态分析（在CTFL®的术语中，这两种技术被称为“静态测试”技术）这些验证策略的补充。

测试工程师需要根据验证策略，验证产品是否符合详细软件设计的要求，以及功能性和非功能性的需求。该策略定义了测试工程师提供证据的方式。这样，测试工程师就可以通过将静态和动态测试技术进行不同形式的组合，实现对单元的验证。

如果开发人员更改了单元，那么测试工程师也必须对此变更进行评估。因此，单元验证的策略也包含回归策略，具体包括：验证更改的代码、确认测试以及重复验证未更改的部分（静态和动态回归测试）。

在SWE.4（软件单元验证）.BP.2（制定单元验证准则）中，ASPICE要求制定单元验证准则，并明确要实现的目标。因此，测试工程师应评估以下两点：单元满足非功能性需求的程度，以及单元与详细设计的匹配程度。在进行单元验证时，可依据以下准则：

- 单元测试用例（包括测试数据）。
- 测试覆盖度目标（例如，判定覆盖）。
- 借助工具进行静态分析，从而评估产品是否符合编码标准（例如MISRA-C，请参阅4.1.1）。
- 对于无法通过工具进行静态分析的单元或部分单元，应进行代码评审。

根据Automotive SPICE（ASPICE），将验证策略文档化是单元级别测试计划（[13]第6.2.7段）的一部分。根据ISO/IEC/IEEE 29119-3，可对文档内容进行拆分，并增加关于静态测试的内容。

2.1.2.6 Automotive SPICE（ASPICE）中的可追溯性要求

与CTFL®核心课程大纲[21]相同，ASPICE也要求具有双向可追溯性[16]。根据双向可追溯性，测试工程师才有可能完成以下工作：

- 对影响进行分析。
- 对覆盖进行评估，或
- 对状态进行跟踪。

而且，测试工程师还可以确保相互关联的元素在内容和语义上保持一致。

[15] 对于术语“验证策略”和“测试策略”，在ASPICE中，我们使用术语“策略”，而不使用ISTQB®中的表达-项目专用“技术”。
[16] 在以下内容中，可追溯性这个术语将始终表示双向可追溯性。

中国软件测试认证委员会（CSTQB®）

ASPICE对纵向/垂直可追溯性和横向/水平可追溯性进行了区分 [9]：

垂直可追溯性：ASPICE要求将利益相关方的需求与软件组件相链接。这样，横跨所有开发级别的链接可确保相关工作产品之间的一致性。

水平可追溯性：ASPICE还要求开发的工作结果与相应的测试规格说明、测试结果之间具有可追溯性和一致性。

此外，基本实践SUP.10（变更请求管理）.BP8（建立双向可追溯性）也要求变更请求和受变更请求影响的工作产品之间具有双向可追溯性。变更请求的发起是因为缺陷或问题的出现，因此，变更请求和相应的问题报告之间也需要具有双向可追溯性。由于常常会有大量和复杂的关联性，因此连续和具有追溯功能的工具链会大有帮助，通过工具链，测试工程师能够高效地创建和管理关联/追溯性。

2.2 ISO 26262（K3）[125分钟]

2.2.1 功能安全和安全文化（K2）[20分钟]

2.2.1.1 E/E系统的功能安全目标

嵌入式系统的功能和技术的复杂性在不断增加。同时，功能强大的、基于软件的电子和电气系统也使得很多新的复杂功能成为了可能，例如汽车的自动驾驶功能等。

基于系统的高复杂性，开发期间的错误行为导致的风险也会随之增加。结果可能是系统处于（未检测到的）故障状态。由于系统存在固有的潜在安全风险，因此安全负责人员就需要对这些潜在风险进行分析。如果确实存在风险，安全负责人员要采取适当的措施，将风险可能带来的影响减小到可接受的程度。

功能安全标准中总结了执行此类分析的方法。基本标准是IEC 61508。国际标准化组织（ISO）根据该标准编制了ISO 26262。

在ISO 26262定义的E/E系统功能安全是指当这些系统发生错误行为（失效）时，避免对生命和肢体造成不可容忍的风险。从这个意义上讲，应将该术语与其他安全术语（例如，信息安全、产品安全或工作安全）予以区分[ISO 26262] [IEC 61508]。工作环境安全和网络安全不在ISO 26262的关注范围之内。如果无法保证网络安全，可能会危及功能安全；如果网络安全得到保证，则有助于产品安全。

中国软件测试认证委员会（CSTQB®）

2.2.1.2 测试工程师对安全文化的贡献

根据ISO 26262，在产品开发过程中，不仅要监督组织的开发过程，而且所有参与者都需要采用独立于过程的方法。每个人都必须理解自己对开发过程和最终产品的安全性造成的影响。外部合作伙伴和供应商也是如此。

参与人员必须了解自己的行为与其他过程紧密相关。开发过程中的每个步骤都会影响功能安全相关需求的遵守和实施情况。这项职责不应随着产品上市而停止，应该一直继续，直到整个系统生命周期结束为止。

测试工程师应认真参与软件开发生命周期的所有阶段，并在过程中不断了解产品开发的整体环境，从而为安全文化做出贡献。[ISO 26262]

2.2.2 测试工程师融入安全生命周期（K2）[15分钟]

安全生命周期描述了以安全为中心的产品开发过程的各个阶段。它从产品理念开始，研究可能存在的安全风险。在根据确定的安全需求制定了规格说明后，再实施到具体产品中。在产品寿命末端产品被处理后，安全生命周期随之结束（另请参阅第1.3节）。

根据ISO 26262，安全生命周期分为以下阶段：

- 第一阶段：产品概念。
- 第二阶段：产品开发。
- 第三阶段：产品生产和维护（在“生产发布”之后）。

供应商的测试工程师工作主要在前两个阶段。在第三阶段中对产品进行变更会导致返回到第一阶段或第二阶段，具体取决于变更涉及的范围。因此，测试工程师也参与修改工作。测试工程师根据安全相关需求（请参阅第2.2.4节）设计测试用例，并选择测试技术在产品开发过程中进行验证，并对需求进行确认。然后，测试工程师将在产品开发过程中的相关子阶段执行这些验证和/或确认。

测试计划活动通常在概念阶段进行。但是，在任何阶段都可以对生成的文档进行调整（例如，关于测试计划或测试规格说明的文档）。测试执行活动主要发生在产品开发过程中不同子阶段切换的时候，比如，在软件实现和软件集成之间，以及进一步到硬件软件集成之时。此外，测试工程师的测试活动对于过渡到第三阶段也具有重要作用[ISO 26262]。

中国软件测试认证委员会（CSTQB®）

2.2.3 ISO 26262结构和与测试相关的部分（K1）[10分钟]

2.2.3.1 标准的设计和结构[扩展知识]

ISO 26262包含10个卷（部分）：

- 词汇（第1卷）。
- 功能安全管理（第2卷）。
- 安全生命周期的各个阶段：
  - 概念阶段（第3卷）。
  - 整个系统、硬件和软件的产品开发（第4-6卷）。
  - 生产和运营（第7卷）。
- 支持过程（第8卷）。
- ASIL和以安全为导向的分析（第9卷）。
- ISO 26262应用指南（第10卷）。

除第1卷和第10卷外，其他每卷都包含结构化的内容。具体包括：

- 概述。
- 适用范围。
- 参考标准和
- 标准依从性要求。

随后是相应卷的具体阐述。每卷中内容的组织结构都相同。比如，在所有卷（部分）中，要执行的活动所采用的结构如下[ISO 26262]：

- 目标。
- 概述。
- 简介。
- 前提条件。
- 更多支持信息。
- 要求和建议。
- 工作结果。

中国软件测试认证委员会（CSTQB®）

2.2.3.2 与测试工程师有关的卷（部分）

对于软件测试工程师来说，软件验证和（至少部分）系统验证至关重要。除第1卷（术语）外，其他几卷（部分）也十分重要：第4卷和第6卷提供了详细的软件验证的建议措施和要求。在选择、设计、实施以及执行相应验证措施时，可以参考这几部分内容。

这些卷重点介绍了系统（第4卷，包括系统确认）和软件级别（第6卷）的测试和验证方面的内容。如果工作中涉及到硬件特定方面，那么测试工程师可以在第5卷中找到相关内容。既与硬件又与软件相关的内容属于软硬件接口的范畴（第4、5和6卷）。

ISO 26262中的第8卷内容比较特殊，介绍了验证过程（涵盖了不同的测试级别）的特性。此外，此卷还包含了对测试工程师非常重要的一些支持过程的要求，例如文档编写和工具的认证。[ISO 26262]

2.2.4 危害程度对测试范围的影响（K2）[20分钟]

2.2.4.1 ASIL的危害程度级别

ASIL（“Automotive Safety Integrity Level/汽车安全完整性等级”）通过采取功能安全的措施，把风险降低到可接受的程度。措施包括，采取一种独立安全功能来监督E/E系统，或实施特定的方法。对于更高级别的风险，必须采取更周密的措施。

在项目开始时，专家团队会对产品进行危害分析和风险评估。对于通过此分析确定的每一种风险，都需要根据标准中提供的某种方法来确定ASIL等级。然后会拟定安全目标和安全需求。在拟定的过程中，会以所确定的ASIL等级为依据。

ISO 26262定义了四个ASIL等级：从ASIL A（最低安全等级要求）到ASIL D（最高安全等级要求）。

如果在经过危害分析和风险评估后，确定的安全需求低于ASIL A的要求，那么根据标准，这些需求与安全性无关，将通过现有质量管理（QM）满足这些要求。[ISO 26262]

2.2.4.2 ASIL对测试技术、测试类型和测试范围的影响

确定的ASIL等级会直接影响测试工程师所执行的测试范围。根据ASIL的特定等级，ISO 26262标准建议执行不同的措施或一系列措施。一般规则是，ASIL等级越高，标准建议采取的措施越广泛、越详细。如果ASIL等级较低，标准通常会提供几个指定的措施以供选择。

ISO 26262规定了三个推荐级别：不推荐、推荐和强烈推荐。“不推荐”表示该标准既不推荐，也不反对采取相应的措施。因此，这些措施可以放心使用。不过，即使执行了这项措施，也仍然需要执行ISO推荐或强烈推荐的措施。

中国软件测试认证委员会（CSTQB®）

对于测试工程师而言，这意味着对于与功能安全相关的系统，应根据ASIL级别使用该标准规定的测试设计技术和测试类型。对于要选择标准提供的哪项建议，测试工程师只能在标准规定的框架内自行决定。例如，对于ASIL A，标准建议使用等价类划分和边界值分析。而对于ASIL B或更高等级，标准则强烈建议使用这些技术（另见第2.2.5节）。

ASIL不是整个产品的一个特性。它与特定的安全目标和由此产生的安全需求相关。因此，如果同一种产品具有不同ASIL等级的安全要求，那么测试的代价可能截然不同。测试工程师在规划测试范围时，必须考虑这一点。[ISO 26262]

2.2.5 CTFL®内容在ISO 26262环境中的应用（K3）[60分钟]

ISO 26262以测试方法列表的形式向测试工程师提供了一些特定的建议，以供其在测试活动中采用。这些表位于第4、5、6和第8卷（部分）中。除了为开发活动和过程提供特定于功能安全的建议外，这些表还提供了可供测试工程师使用的技术。

在这些表中，对于所有适用的与技术或活动相关的建议，该标准均使用术语“方法”以作表述。在这方面，功能安全术语与ISTQB®中的术语略有不同。对于测试工程师来说，ISO 26262中的以下方法尤为重要：

- 测试设计技术（例如，等价类划分、边界值分析等）。
- 测试执行所使用的技术（例如，部件或系统的仿真、原型）。
- 测试类型（例如，性能测试、渗入测试等非功能性测试）。
- 测试环境（例如，HiL、车辆等）。
- 静态测试技术（例如，评审、静态分析等）。

这些测试方法列表推荐了针对每个ASIL等级的测试方法。

这些表的设计结构始终保持一致：

		ASIL A	ASIL B	ASIL C	ASIL D
1	方法x	o	+	++	++
2	方法y	o	o	+	+
3a	方法z1	+	++	++	++
3b	方法z2	++	+	o	o

表1：测试方法列表示例

中国软件测试认证委员会（CSTQB®）

对于每种方法，会根据ASIL等级，将其推荐级别记录下来，例如，是推荐（+）还是强烈推荐（++）。对于标记为可选（o）的方法，表示标准既不建议使用，也不反对使用。

ISO 26262还在表格中提到了等效替代方法（请参阅上述示例中的第3a和3b行）。此时，测试工程师需要选择合适的组合，以能够符合ASIL的方式检查相关需求。测试工程师应说明选择这种组合的理由。

如果某些方法没有替代方法（请参阅示例中的第1行和第2行），那么也就不存在对方法的组合。此时，理想情况下，测试工程师应根据ASIL等级，采用所有强烈推荐的方法。

在上面的示例中，若要证明需求符合ASIL C，可采用以下方法：

- 方法x：强烈推荐，如果按照ISO 26262进行开发，通常加以运用。
- 方法y：推荐，如果对证据有用，可加以应用。
- 方法z1和z2：针对这种情况，至少应选择方法z1，因为对于ASIL C来说，它的要求等级较高。

ISO 26262允许测试工程师使用表中所列方法以外的其他方法。但是，如果选择其他方法，测试工程师必须说明所选择的这些方法的用途和适用性。[ISO 26262]

中国软件测试认证委员会（CSTQB®）

2.3 AUTOSAR（K1）[15分钟]

简介

AUTOSAR （AUTomotive Open System ARchitecture）是“汽车开放系统架构”的缩写，也代表了它背后的开发联盟。AUTOSAR是在2003年成立的，成员主要包括汽车行业的生产商和供应商。这个联盟的目标是“为在汽车环境中运行的软件架构创建可免费使用的标准”。因此，本标准旨在解决软件日益增加的重要性和复杂性问题[14]。如今，AUTOSAR 已成为全球公认的E/E系统标准。所以，测试工程师必然会接触到AUTOSAR产品。因此，对于测试工程师来说，了解AUTOSAR的目标、基本设计以及其与测试工程师工作之间的关系非常重要。

2.3.1 AUTOSAR的目标（K1）[5分钟]

AUTOSAR要实现的以下项目目标，以“在标准上协作，在实现上竞争”为原则：[14，15]：

1. 支持软件的可转移性（可移植性）。
2. 支持不同车辆和不同平台（可扩展性）。
3. 支持不同的功能域。
4. 定义可维护、可调整和可扩展的开放架构。
5. 支持可靠系统的开发，具体体现在这几方面：可获得性、可靠性、安全性（功能安全以及网络安全，即“安全和保障”）、完整性和可维护性。
6. 支持自然资源的可持续利用。
7. 支持合作伙伴之间相互协作。
8. 实现汽车电控单元（ECU）基础软件功能的标准化。
9. 支持适用的汽车标准和最先进的技术。

2.3.2 AUTOSAR的总体结构（K1）[扩展知识][5分钟]

AUTOSAR的架构由三个独立的层组成：

- 独立于硬件的层，包含AUTOSAR软件组件（SW-C）。
- 以硬件为导向的层，包含标准化的基础软件（BSW）。
- 抽象层，包含AUTOSAR运行时环境（RTE）。该层可控制电控单元内外的数据交换，即在软件组件之间，以及软件组件和基础软件之间实现数据交换。

中国软件测试认证委员会（CSTQB®）

此外，AUTOSAR的架构还包括AUTOSAR方法论，适用于协调控制单元软件的开发。通过此方法论，OEM和供应商可通过AUTOSAR模板（所谓的“arxml文件”）交换有关描述文件的信息。[14，16]：

- “ECU配置描述”中包含用于在电控单元上集成SW-C的数据。
- “系统配置描述”中包含用于集成一辆汽车中所有控制单元的数据。
- “ECU提取”中包含从“系统配置描述”中提取的单个电控单元的数据。

2.3.3 AUTOSAR对测试工程师工作的影响（K1）[5分钟]

AUTOSAR会对测试工程师的工作产生影响，尤其会对以下测试级别产生影响[18]：

- 虚拟环境中（例如，软件在环）的软件组件测试和软件集成测试：借助虚拟BSW和RTE，测试工程师可以尽早测试应用程序的软件组件[17，18]。
- 真实控制单元中的软件测试和软件集成测试：在这些测试中，测试工程师可以访问RTE上的通信。在这种情况下，测试工程师还可以触发和测量SW-C的运行时行为[19]。
- AUTOSAR验收测试是对软件系统进行的合规性测试，可确保软件符合AUTOSAR对通信层和应用层的功能要求。AUTOSAR验收测试的执行是可选的[20，21]。
- 系统集成测试：不同电控单元的功能集成和连接（例如，在整车环境中）。通过模拟缺失的功能（可能是分布式的功能），测试工程师可以尽早对系统行为进行评估[17]。

2.4 比较（K2）[20分钟]

2.4.1 ASPICE的目标和ISO 26262的目标（K1）[5分钟]

有许多不同的标准针对产品开发提出了要求。通常，这些要求分别针对开发过程中的不同方面。此处将ISO 26262的目标和ASPICE的目标进行了比较。

ISO 26262[3]的目标是，通过提出合适的要求和过程，避免在开发过程中出现系统性故障以及在运行过程中出现硬件故障的风险。对于E/E系统的开发，该标准针对测试工程师使用的过程和方法提出了要求。这些要求取决于项目的ASIL等级。

ASPICE[9]的目标是，在一个框架内评定产品开发过程的能力。为此，ASPICE为这些过程定义了评估准则。与ISO 26262相反，这些准则与危害程度和产品ASIL等级无关。

[18] 根据测试级别；另请参阅2.4.2。

中国软件测试认证委员会（CSTQB®）

2.4.2 测试级别比较（K2）[15分钟]

ISO 26262和ASPICE均对测试级别进行了介绍。但是，这些测试级别与CTFL®[21]的测试级别有所不同。因此，为了实现高效合作，测试工程师应该对所有测试级别有一致的理解。

ASPICE中使用的术语“系统”，和ISO 26262中使用的术语“系统”与“相关项”均指由硬件和软件组件组成的产品。但是，CTFL®中的术语“系统”却指软件。因此，可以将ISTQB®[21]中的测试级别与ISO 26262和ASPICE中的测试级别对应起来，如下所示：

ISTQB®	ISO 26262	ASPICE 3.0
验收测试	安全性验证（4-9）[19]	无等效项
综合系统测试[20]	相关项集成和测试（4-8）[21]	系统合格性测试（SYS.5）
系统集成测试		系统集成测试（SYS.4）
系统测试	软件安全性需求验证（6-11）	软件合格性测试（SWE.6）
组件集成测试	软件集成和测试（6-10）	软件集成测试（SWE.5）
组件测试	软件单元测试（6-9）	软件单元验证（SWE.4）

表2：测试级别分配

根据ISTQB® CTFL®核心课程大纲（[21]，[48]），大多数测试技术的适用性与测试级别无关。ASPICE通常也不会为测试级别指定测试技术。因此，测试工程师可自行选择测试技术。相反，在ISO 26262中，每个测试级别都有单独的测试方法列表（请参阅第2.2.5节和第2.2.4.2节）。对于应使用的测试技术，这些方法汇总表根据ASIL等级为测试工程师提供了建议。

中国软件测试认证委员会（CSTQB）

[19] 根据ISTQB®，安全性验证仅覆盖部分验收测试
[20] 对几种异构分布式系统进行测试[34，39]。
[21] 相关项集成和测试包括三个阶段：元素的软硬件的集成和测试、相关项中所有元素的集成和测试，以及该相关项与车辆环境中的其他相关项的集成和测试。

3. 在虚拟环境中进行测试（K3）[160分钟]

关键词

模型在环（Model in the Loop）（MiL），软件在环（Software in the Loop）（SiL），硬件在环（Hardware in the Loop）（HiL），开环系统（Open-Loop-System），闭环系统（Closed-Loop-System），环境模型（汽车）（Environment model）（Automotive）。

学习目标

AUTFL-3.1.1	（K1）	回顾在汽车开发过程中建立测试环境的目的/动机。
AUTFL-3.1.2	（K1）	回顾汽车特定测试环境的常规组件。
AUTFL-3.1.3	（K2）	回顾闭环系统和开环系统之间的区别。
AUTFL-3.1.4	（K1）	回顾汽车控制单元的基本功能、数据库和协议。
AUTFL-3.2.1.1	（K1）	回顾MiL测试环境的结构。
AUTFL-3.2.1.2	（K2）	阐述MiL测试环境的应用范围和边界条件。
AUTFL-3.2.2.1	（K1）	回顾SiL测试环境的结构。
AUTFL-3.2.2.2	（K1）	回顾SiL测试环境的应用范围和边界条件。
AUTFL-3.2.3.1	（K1）	回顾HiL测试环境的结构。
AUTFL-3.2.3.2	（K2）	阐述HiL测试环境的应用范围和边界条件。
AUTFL-3.2.4.1	（K2）	根据XiL测试环境（MiL、SiL和HiL）的区别，总结各自环境下进行测试的优缺点。
AUTFL-3.2.4.2	（K3）	根据一定的准则，分配测试到一个或多个测试环境。
AUTFL-3.2.4.3	（K1）	在V-模型中区分的三个XiL测试环境（MiL、SiL、HiL）。

中国软件测试认证委员会（CSTQB®）

3.1 测试环境概述（K2）[30分钟]

3.1.1 在汽车软件开发过程中建立测试环境的动机（K1）[5分钟]

测试工程师面临着一些特殊的挑战。一方面，测试工程师需要尽早开始测试，以便尽早发现开发过程中的缺陷。另一方面，测试工程师还需要在尽可能真实的环境中测试系统，以找出成品中可能会出现的缺陷。测试工程师可以使用与不同开发阶段相匹配的测试环境来解决此矛盾。为此，测试工程师可以在量产或开发的电控单元（ECU）完全可用之前，实施和执行个别测试任务。测试工程师通过使用不同的测试环境，可以模拟场景并执行在实际车辆环境中难以展现的测试用例，例如，电源线短路和断路，或网络通信过载。[24]

3.1.2 测试环境的常规组件（K1）[5分钟]

测试工程师若要能够执行测试，需要一个可以模拟缺失组件的测试环境。在这个测试环境中，测试工程师可以触发输入并观察输出，也称为“控制点”（PoC）和“观察点”（PoO）。根据ISO/IEC/IEEE 29119，测试环境由以下组件构成：

- 测试环境的硬件（计算机、具备实时处理能力的计算机（如需要）、测试台、开发工具包等）。
- 测试环境软件（操作系统、仿真软件、环境模型）。
- 通信设施（网络入口、数据记录仪）。
- 工具（示波器、测量工具）。
- 实验室（防止电磁辐射和噪音）。

测试环境的一个重要组件是环境模型，该模型也是虚拟测试环境的重要组成部分。它模拟了真实环境的必要元素，例如发动机、变速箱、车辆传感器和电控单元，甚至是驾驶员或道路状况。测试环境还包含不同的访问接口。测试工程师可以使用这些接口来测量和观察被测项[25]。

3.1.3 闭环系统与开环系统的区别（K2）[15分钟]

测试环境触发测试项的输入，并监督其输出（通过输出接口）。然后，对输出接口的行为进行分析。如果测试成功，观察到的行为会与预期的输出相符。

通常，有两类控制系统：闭环系统和开环系统。两者之间的差异在于电控单元对其环境的反应方式，这种反应方式的不同会导致对虚拟测试环境的仿真需求不同。

中国软件测试认证委员会（CSTQB®）

3.1.3.1 开环系统

在开环系统中，系统输入与输出无关。系统处于开环状态，没有任何反馈。在这种情况下，将由测试工程师直接在测试规程中定义测试项的输入。

开环系统和闭环系统的应用场景在很大程度上取决于测试项的工作原理。如果测试项具有应对行为或者是反映一个状态机，则应首选开环系统。在内饰和底盘电子系统中，有许多开环系统的示例（请参考照明灯和开关）。

3.1.3.2 闭环系统

闭环系统（也称为在环系统）中的激励（输入）会考虑测试项目的输出。这是通过环境模型来完成的，环境模型会收集测试项的输出并将其直接或间接地转变成测试项的输入。因此，在测试环境中形成了一个控制闭环。

对控制器进行测试的过程中，更多的是使用闭环系统。因为使用闭环系统，测试工程师不仅可以测试复杂的功能，如发动机和变速箱的控制，还可以测试驾驶员辅助系统，如防抱死制动系统（ABS®）或车辆稳定性控制系统（ESP®）。[26，27]

3.1.4 电控单元的基本接口、数据库和通信协议（K1）[5分钟]

汽车环境中的控制单元是一种嵌入式系统，由硬件和软件组成。电控单元接收不同的模拟输入和数字输入，即以电压、电流和温度的形式不断收集环境数据。而且，通信总线系统会向控制单元提供进一步的信息。这些信息来自智能传感器或其他电控单元（这些电控单元或智能传感器也会自行收集、处理信息或生成信息）。测试对象管理内存中的数据，并决定输出动作、信息或数据。这些输出也是通过模拟和数字输出引脚、总线系统或诊断接口来传出的。

数据库是存储数据的仓库，定义了控制单元的输入和输出信号。这些数据还包括信号的描述、单位和转换公式。

通信协议描述了通过相应物理接口进行的数据交换。这些协议定义了哪些电压或位序列代表哪些信号值。

数据库和通信协议的选择取决于电控单元的功能。例如，要访问控制单元中的诊断功能，测试工程师需要所用数据库的相关信息（例如ASAM MCD2 D；以及“开放式诊断数据交换”）和通信协议（ISO 14229中的“统一诊断服务”）。ASAM标准[27，28]中定义了更多特定于汽车的数据库。

中国软件测试认证委员会（CSTQB®）

3.2 在XiL测试环境中进行测试（K3）[130分钟]

在汽车行业中，会使用以下类型的XiL测试环境：

- 模型在环（MiL）。
- 软件在环（SiL）。
- 处理器在环[22]（PiL）。
- 硬件在环（HiL）。
- 车辆在环[23]（ViL）。

测试工程师应熟悉并理解这些测试环境（MiL、SiL和HiL）。以下几段深入探讨了不同测试环境的结构和应用范围。从这个意义上讲，XiL是代表这些不同测试环境的通用术语。

3.2.1 模型在环（MiL）（K2）[20分钟]

3.2.1.1 MiL测试环境的结构

在MiL测试环境中，测试项是以模型的形式存在的。此模型是可执行的，但并没有针对一款硬件进行编译。这些模型由开发人员使用专门的建模工具构建而成。测试工程师如果想要执行和测试这些模型，需要搭建测试环境。而测试环境常常也是在与测试项本身所处的同一开发环境中创建的。该测试环境还可以包含环境模型。测试工程师可以通过访问接口来触发和观察测试项。访问接口既可以放在测试项的模型中，也可以放在环境模型中。测试项的模型与环境模型直接相连，因而能够轻松实施并用作闭环系统。

3.2.1.2 MiL测试环境的应用范围和边界条件

在MiL测试环境中，测试工程师能够测试功能性系统设计。在开发过程中（遵循通用的V-模型），测试工程师既可以测试单个组件，也可以测试整个系统。若要执行此测试，测试工程师需要一台计算机和相应的仿真软件，包括环境模型。随着测试项的功能范围不断扩大，环境模型也随之变得更加复杂。再加上实际情况和环境因素本身也十分复杂，从而导致模型的执行时间会不成比例地增加。因此，从开发的某一特定阶段开始，已无需再花费时间实施MiL测试环境[24]。

通过使用MiL测试环境，测试工程师可以在开发的早期阶段（即V-模型的左侧）测试模型在所有开发级别的功能。但是，在环境模型中仿真总线、诊断功能或物理行为（例如断路或短路）的方式并不常见。相反，在其他测试环境中，可以更轻松并且以更低的成本执行这些任务。

[22] 这类测试环境在本课程大纲中不予考虑，仅属于扩展知识。
[23] 这类测试环境在本课程大纲中不予考虑，仅属于扩展知识。
[24] 所有其他XiL测试环境均是如此。

中国软件测试认证委员会（CSTQB®）

在MiL测试环境中，必须认识到测试执行并不是实时的。由于所有组件都可以作为模型的形式提供，因此会按照模型的仿真时间执行测试。系统越复杂，为了提供模型仿真所需的数据，所需计算机的计算时间就越长，或需要功率更大的计算机才能胜任。在较小的系统中，仿真所用的时间短于实时执行的时间。但是，这也有一大优势，就是测试工程师可以随时暂停仿真过程以进行详细的分析和评估。

3.2.2 软件在环（SiL）（K1）[10分钟]

3.2.2.1 SiL测试环境的结构

测试项已经针对特定的SiL测试环境进行了编译。这表示已使用软件工具为特定计算机架构编译了源代码。这些编译后得到的机器码由二进制数据集组成，因此只能由测试环境读取。如果希望测试环境能够访问信号，需要使用封装器。封装器是一个附加软件，可为机器码创建特定的访问接口。这样，测试工程师就可以触发软件信号并加以观察。封装器定义测试项的访问接口，但不会执行其功能性任务。

要进行仿真，需要使用环境模型。通过封装器，可将测试项连接到测试环境。测试在计算机上执行，无需特定的硬件。测试工程师需要一个软件工具，该工具能够为测试项创建一个封装器，其中包含能够访问测试环境的接口。

3.2.2.2 SiL测试环境的应用范围和边界条件

如果开发人员基于模型生成源代码，则软件的实际行为可能与预期行为有所不同。导致这种差异的原因是：模型中的数据类型（主要是浮点类型）和编译后软件代码中的数据类型（主要是定点类型）不同，以及二者的内存空间不同。预期行为中的这些偏差可以在SiL测试环境中进行首次测试。测试工程师可以使用诸如背靠背测试（另见4.2.2）之类的技术对这些行为进行比较。

与MiL测试环境类似，SiL也是按照仿真时间进行测试的。仿真时间可能会长于或短于实时时间，具体取决于（积分）计算方法和环境模型的复杂程度。测试工程师可以随时暂停仿真操作以进行详细的分析和评估。功能测试、接口测试和回归测试是很常见的可以在SiL测试环境中实施的测试类型。而性能测试和可靠性测试则不太经常在SiL环境中实施，这些软件特性主要受目标硬件的影响。

3.2.3 硬件在环（HiL）（K2）[20分钟]

3.2.3.1 HiL测试环境的结构

如果测试对象可提供原型或已完成开发，则测试工程师可以在HiL测试环境中执行测试。HiL测试环境通常由以下部分组成：

中国软件测试认证委员会（CSTQB®）

- 可提供不同电压的电源。
- 用于运行环境模型的、具备实时处理能力的计算机。
- 其他未在环境模型中仿真的真实部件。
- 信号类型和幅值的处理。
- 用于仿真断路和短路的故障注入单元（FIU，另请参阅4.2.3）。
- 串接于线束中的断线测试盒。
- 剩余总线仿真：模拟不存在的总线节点。

3.2.3.2 HiL测试环境的应用范围和边界条件

HiL测试环境中具有多种接口。测试工程师必须注意，使用错误的接口会导致测试结果无效。知晓HiL测试环境中不同的接口及其连接方式，测试工程师可以有效地实施、执行和评估测试。

与前面提到的测试环境（MiL和SiL）相比，HiL测试环境更为复杂，因为它由多个部分组成。测试工程师必须掌握这一复杂性，才能胜任测试任务。HiL测试环境可用于组件测试、集成测试和系统测试。其目的是发现软件和硬件中的功能性和非功能性缺陷以及其他问题。

在HiL测试环境中，可以对不同的测试级别进行分析。如果测试项是单个电控单元（ECU），称为组件[25]HiL。如果测试项是多个电控单元的组合，称为系统HiL。测试工程师可以使用组件HiL来测试控制单元的功能。系统HiL则侧重于对电控单元之间的数据交换进行测试，以及对整个系统进行系统测试。

与前面提到的测试环境（MiL和SiL）不同的是，HiL测试环境中的仿真时间与实时时间始终相同，原因在于软件是在硬件上实时运行的。在此测试环境中，无法暂停或停止仿真操作。因此，这种测试环境包含具备实时处理能力的计算机，以便能够在预定的时间段内收集和处理所有相关信号。

中国软件测试认证委员会（CSTQB®）

[25] 在这种情况下，术语“组件”指E/E系统环境中的电控单元（ECU）。

3.2.4 XiL测试环境的比较（K3）[80分钟]

3.2.4.1 在XiL测试环境中进行测试的优缺点

测试工程师应了解不同测试环境的特点，这样才能了解并评估在每种测试环境中进行测试的的优缺点。比较标准和结果如表3所示。

标准	MiL测试环境	SiL测试环境	HiL测试环境
与实际情况的接近程度	低	低到中	高
	实际情况是仿真的，其中许多特性是抽象的，重点在于结构和逻辑的测试	编译后的真实软件（不包含硬件）可以执行	集成系统，能够运行
调试所用时间及工作量	低	中	高
	发现测试项的模型的缺陷（模型调整）	发现软件代码中的缺陷（软件调整）	发现系统级别的缺陷（系统调整）
实施和维护的工作量	低	中	高
	创建环境模型	创建环境模型和封装器	创建环境模型并连接硬件组件
测试准备工作的工作量	低	中	高
	可以快速建立环境	可以快速建立环境	测试的设计、实施和评估工作量较大
测试项的成熟度要求	低	中	高
	系统的模型是仿真的	初始功能可在目标软件中进行测试	可对一个或多个可执行的电控单元或系统部分进行尽可能完整的测试
测试依据（规格说明）的详细程度要求	中	中到高	高
	如果没有完整的规格说明，则模型将被用于测试，甚至模型能够对确定规格说明发挥部分作用	必须提供关于软件级别的相关信息（详细的组件规格说明）	具备系统级别的、可测试的需求（完整的系统规格说明）
测试项的可访问性	高	中	低
	可以观察和控制模型中的所有信号。	只能观察和控制封装器中的信号。	只能观察和控制硬件上的信号或通信协议中的信号。

表3：比较标准及其对MiL、SiL和HiL测试环境的影响

中国软件测试认证委员会（CSTQB®）

3.2.4.2 将测试用例分配给一个或多个测试环境

下表中详细描述了测试目标，并将这些目标分配给了合适的测试环境。

测试类型	描述（通过示例）	MiL	SiL	HiL
测试客户需求	正确交付客户所需的功能，包括正确处理输入，对输入进行正确的反应，以及在出口点的正确输出数据。	0	0	+
对于缺陷检测和处理的测试机制	• 硬件随机故障的检测和处理 • 软件缺陷的检测和处理 • 当检测到缺陷后，切换到安全状态- 例如，禁止该系统	+	+	+
测试对配置数据的反应	检查配置数据（例如参数集或变体编码）对测试对象行为的影响。	0	+	+
测试诊断功能	正确交付所需的诊断功能，如缺陷检测、缺陷设置和重置需求，以及在诊断存储中设置故障码（例如，车载诊断或在车库中进行诊断）	-	+	+
测试接口上的交互	检查测试项的内部和外部接口	0	+	+
证明易用性	测试对象运行的表现应该与要求和用户的预期一致。	-	0	+
提示：+表示推荐，0表示可能，-表示不可取				

表4：比较在MiL、SiL和HiL测试环境中的测试类型

此表说明这些测试环境可能适用于某些特定的测试目标。这种灵活多样的方法尤其在缺陷检测和处理机制方面表现得明显。根据“测试前置Front-Loading”[26]原则，我们得出了一个一般性结论，即通过测试能够尽早检测基本需求，发现设计缺陷。因此，MiL用于检测总体设计缺陷，SiL主要用于检测软件缺陷，HiL用于检测硬件/软件缺陷。而且，需要注意的是，除了稳定性、可靠性、效率、性能以及易用性这些方面之外，所有测试类型都将测试项的功能适用性作为重点测试对象。

在测试策略中，测试工程师（作为测试经理角色）会将测试范围分配给多个不同的测试环境。通过综合考虑表3和表4中的标准，测试经理可以选择最佳的测试环境。

3.2.4.3 XiL测试环境（MiL、SiL、HiL）在通用V-模型中的位置

技术性的系统设计位于V-模型的左侧。测试工程师可以在MiL测试环境中测试此设计。如果进一步开发测试项和MiL测试环境，测试工程师还可以在此测试环境中执行组件和集成测试。

在测试对象的单个组件完成代码编译后，测试工程师可以使用SiL测试环境。在SiL测试环境中可执行的典型测试是组件测试和集成测试。这些测试类型位于V-模型的右侧。

在系统测试中，由于测试项的特定功能已完全开发，因此测试工程师可以在HiL测试环境中执行

[26] 越早检测到缺陷越好。

中国软件测试认证委员会（CSTQB®）

系统测试。[24]

通过将测试环境正确分配到测试级别，可以根据以下三个方面来优化整个测试过程：

产品风险最小化

- 找出每个测试级别中特有的故障类型（例如，在HiL环境中执行系统级别的性能测试）。

测试成本最小化

- 对于每种测试类型，选择合乎要求的测试级别。
- 将测试移到早期低成本和虚拟测试级别。

标准依从性

- 在ISO 26262标准的测试方法列表中，建议根据ASIL等级使用相应的测试环境。

中国软件测试认证委员会（CSTQB®）

4. 专门用于汽车行业的静态和动态测试技术 [230分钟]

关键词

编码标准（Coding standard），背靠背测试（back-to-back testing）。

学习目标

静态测试技术

AUTFL-4.1.1	（K2）	举例说明MISRA-C:2012指南的目的和要求。
AUTFL-4.1.2	（K3）	使用ISO/IEC 29148标准中与测试工程师相关的质量特性对需求进行评审。

动态测试技术

AUTFL-4.2.1	（K3）	创建测试用例以实现MC/DC测试覆盖。
AUTFL-4.2.2	（K2）	举例说明背靠背测试的用法。
AUTFL-4.2.3	（K2）	举例说明故障注入测试的原则。
AUTFL-4.2.4	（K1）	回顾基于需求的测试原则。
AUTFL-4.2.5	（K3）	根据测试需要依赖于测试对象的这一准则，来选择所需的合适测试设计技术。

中国软件测试认证委员会（CSTQB®）

4.1 静态测试技术（K3）[75分钟]

简介

静态测试是指不运行软件开发过程中的工作产品而对其进行检查的一种测试方法。具体包括人员评估（评审）和借助工具进行的静态分析。

4.1.1 MISRA-C:2012指南（K2）[15分钟]

这是目前开发人员在编程时遵循的最先进的指南之一。ISO 26262标准也建议对安全相关的软件使用该指南[27]。这些编码标准有助于避免软件中出现异常，而这些异常可能会导致软件中出现缺陷。同时，这些标准还有助于开发人员提高软件的可维护性和可移植性。

MISRA-C:2012指南[15]中包含C语言的编程指南。它定义了两种类型的指南：

- 可通过静态分析工具进行验证的规则。例如，源代码不包含嵌套注释。
- 无法通过静态分析工具完全验证的指令。其原因在于，这些指令更多的涉及开发的过程和软件的外部文档。例如，开发人员是否完整地记录了其实现的功能（行为）。

每条指南都被归类为以下三个级别中的一个：

- 建议：在工作量允许的情况下，开发人员应该遵守。
- 要求：只在开发人员可以确切地阐述不遵守这项规则的原因的情况下，才可以忽略
- 强制：开发人员必须遵循。无一例外。

组织可以单独强化一条规则或一条指令的要求，但不能降低其要求。

4.1.2 需求评审的质量特性（K3）[60分钟]

规格说明是开发和测试的依据。因此，如果这些规格说明中存在缺陷，则会导致后期活动的成本和时间急剧增加。如果在开发后期阶段（例如验收测试或运行中）才检测到缺陷，这种影响尤其明显。评审是一种有效的措施，可以尽早发现规格说明中的缺陷，从而以较少的成本尽早修复缺陷。

在测试分析阶段，测试工程师必须检查测试对象的规格说明[21]。尤其应检查规格说明是否适合作为测试依据。质量特性可帮助测试工程师在评审规格说明的过程中聚焦关键点，并发现尽可能多的缺陷。ISO/IEC/IEEE 29148:2011[37]中包含单一需求的质量特性，以及需求组群的质量特性。

[27] 另请参阅[ISO 26262:2011]第9部分的表6。

中国软件测试认证委员会（CSTQB®）

ISO/IEC/IEEE 29148:2011中与测试工程师相关的需求特性

一组需求中单个需求的特性：

- 可验证性：每个需求都可以通过静态或动态测试来进行验证。
- 明确性：每个需求都包含明确的测试条件。
- 一致性：每个需求内部以及它与其他需求之间都保持一致。
- 完整性：每个需求都要考虑所有可能的情况（包括错误、中止和异常场景）。同时，对使用的所有图表都进行了标注；定义了缩写和术语的含义。
- 可追溯性：每个需求都有明确标记（例如，通过ID）。这是可以进行影响分析的前提，并且测试用例的覆盖度也是透明和明了的。
- 设定了边界（针对一组需求）：明确定义了开发和测试的范围。
- 原子性：无法再将需求进一步细分成有意义的更小部分。

测试工程师可以根据各项特性制定一个检查表，以作为评审的工具。这些评审检查表包括一些针对前面所述内容的问题。测试工程师必须竭尽全力进行回答。下面列出了一些针对每个需求可能会提出的测试工程师必须予以回答的问题：

- 可验证性：是否可以通过相应测试级别的静态或动态测试来验证该需求？
- 明确性：需求是否定义明确，不留变通解释的余地，或者需求没有基于隐性知识或经验知识？
- 一致性：需求内部以及它与其他需求之间是否均保持一致？
- 原子性：需求是否无法被进一步细分，例如，是否可以分解需求中诸如if-then-else这类构造中的逻辑关系而得到细分的需求？

霍布斯认为：对于安全关键系统的嵌入式软件开发[30]，需求还应具有以下特性：可行性、不受限于具体的实现、必要性。测试工程师通常难以对这些特性进行评估；但是，这些特性会对测试设计造成一定程度的影响。

4.2 动态测试技术（K3）[155分钟]

4.2.1 条件覆盖测试、条件组合覆盖测试、MC/DC覆盖测试（K3）[60分钟]

此处描述的技术是白盒测试设计技术的一部分（更多详情，请参阅课程大纲 CTAL-TTA）。测试工程师可以直接从测试项的结构（例如，源代码）中推演测试用例。

中国软件测试认证委员会（CSTQB®）

相对于判定测试，即测试工程师可以根据代码中的判定覆盖设计测试用例（请参阅[21]），条件测试指的是满足判定中的单个条件。因此，这些技术面向如何进行判定的问题：每个判定由一个或多个“原子”条件组成。如果测试工程师执行测试用例，那么这些判定中的每个条件的值可以是“true/真”或“false/假”。然后，判定的总值来自这些单个值[7]的逻辑组合。

如果判定只包含一个条件，那么这些技术与判定测试完全相同。否则，这些技术将有所不同，如下所示[7]：

- （简单）条件测试（表5中的技术A）：测试工程师设计测试用例的目的是覆盖每个条件的true/false结果。如果选择的测试数据不合适（见表5），虽然可以实现100%（简单）的条件覆盖，但不能完全覆盖判定结果。比如在下表中，单个条件B1和B2分别被赋值为true和false时，它们的的判定结果均为“false”。
- 条件组合测试（下表中的技术 B）：测试工程师设计这些测试用例的目的是覆盖所有单个条件的值的所有组合。如果值的所有组合都经过了测试，那么自然每个判定结果也经过了测试。
- 修正条件/判定覆盖测试（MC/DC覆盖测试）（下表中的技术C）：此类测试类似于条件组合测试（B）。不过，该技术仅考虑特殊的组合条件，即组合中的各个条件（B1，B2）会单独影响判定结果。在测试用例TC4中，将B1或B2从“false”改为“true”不会导致判定结果发生变化（即，仍为“false”）。因此，通过TC1、TC2和TC3就可实现100% MC/DC覆盖；无需考虑TC4。

表5通过示例，说明了根据所选测试技术，要实现100%覆盖而需要的测试用例：

	单个条件		表达式的判定结果：E=B1 AND B2	技术		
测试用例	B1	B2		A	B	C
TC1	B1=TRUE	B2=FALSE	E=FALSE	X	X	X
TC2	B1=FALSE	B2=TRUE	E=FALSE	X	X	X
TC3	B1=TRUE	B2=TRUE	E=TRUE		X	X
TC4	B1=FALSE	B2=FALSE	E=FALSE		X	

表5：条件测试（A）、条件组合测试（B）和修订条件/判定覆盖测试（MC/DC覆盖测试）（C）的技术比较

该示例还显示了这些技术的局限性：如果选择（简单）条件测试（A），尽管会实现100%的条件覆盖，但测试工程师仍面临仅覆盖一个判定结果的风险。选择合适的测试用例可以避免这种情况（例如选择TC3和TC4）。

中国软件测试认证委员会（CSTQB®）

通过使用条件组合测试（B），测试工程师可以覆盖所有可能的输入和输出。但是，要测试的用例数量是这几项技术中最多的。

通过使用MC/DC覆盖测试（C），测试工程师使用比条件组合测试更少的测试用例，就能实现所有单个条件和所有判定的完全覆盖。

4.2.2 背靠背测试（K2）[15分钟]

背靠背测试（也称：比较测试[32]）更像是一种测试方法，而不是一种测试（设计）技术。这种方法是将测试项的两个或多个变体进行比较。为此，测试工程师需要对所有变体执行相同的测试用例并比较结果。如果结果相同，表示通过测试。如果结果不同，需要分析检测到差异的原因。

从内容的角度来看，测试项必须基于相同的需求。因为只有这样，才能使行为具备可比较性。这些需求不会作为测试设计的测试依据。相反，背靠背测试会显示测试项或测试环境之间最细微的差异。因此，此类测试不可取代基于需求的测试。

最简单的一种情况是，背靠背测试将同一软件的不同版本作为测试项。例如，在这种情况下，可将较低版本的测试项作为背靠背测试的测试准则（类似于回归测试）[33]。另一种情况是将可执行模型与（手动或自动）生成的代码[32]进行比较。这是一种基于模型的测试，在这类测试中，也可以将可执行模型作为测试准则[34]。因此，这项技术非常适合自动化测试设计。通过这项技术，测试工程师不仅可以从模型中获得预期的结果，而且还能获取自动化测试用例。

4.2.3 故障注入测试（K2）[15分钟]

故障注入测试更多地是一种健壮性测试方法，而非一种特殊测试（设计）技术。像错误处理之类的编程技术，其目的是使系统以稳妥安全的方式对内部和外部缺陷做出反应。要对这些编程技术进行测试，测试工程师可以选择性地将以下几类缺陷[34]注入到系统中：

- 外部组件中的缺陷：例如，系统需要安全地检测来自传感器的不可信值。
- 接口中的缺陷：例如，系统功能不能因短路或丢失报文而受到影响。
- 软件中的缺陷：系统应该检测并处理内部缺陷。

在经典故障注入测试中，测试工程师可以通过操控真实的组件来输入缺陷。

测试工程师可以在程序运行的时候模拟外部缺陷（以及接口缺陷）。比如，在HiL测试环境中，执行故障注入测试。在这里，故障注入单元（FIU）[35]是真实部件缺陷的仿真器。在这些缺陷中，短路和断路缺陷尤为常见。而在SiL测试环境中，软件接口的缺陷可以被仿真。

对于软件中的缺陷通常只能在开发环境中注入，例如通过调试器或XCP。因此，实际的执行过程往往非常耗时。

中国软件测试认证委员会（CSTQB®）

4.2.4 基于需求的测试（K1）[5分钟]

基于需求的测试更多地是一种测试方法（实践）[22]，而不是一种测试（设计）技术。这种测试方法旨在通过测试用例来覆盖需求。这样，测试工程师就能确定测试项是否满足需求。

在此方法中，测试工程师会分析需求、推导出测试条件、设计测试用例并执行这些测试用例。测试工程师通过对测试结果的分析，可以改进测试用例。为此，测试工程师还可以创建更多的测试用例，并应用更多的测试实践（例如，基于经验的测试）。因此，测试工程师通过回归测试（以探索性测试的形式）这样的测试，可以降低缺陷带来的风险。

如果需求不完整或不一致，那么以此为依据而设计的测试会出现相同的问题。另一方面，如果需求过于详细，那么测试工程师可能无法测试所有需求。因此，必须对测试用例进行优先级排序[3]。

4.2.5 根据环境选择测试技术（K3）[60分钟]

ISO 26262标准（第6卷）建议测试工程师应根据ASIL等级来应用测试设计技术（请参阅第2.2章）。除了其他一些技术之外，这些技术还包括在CTFL®中和之前在第4.2章中提到的以下技术：

- 基于需求的测试。
- 等价类划分。
- 边界值分析。
- 语句测试。
- 判定测试。
- MC/DC覆盖测试。
- 错误推测。
- 故障注入。
- 背靠背测试。

然而，具体决定使用何种技术，取决于包括以下内容在内的一些因素：

技术发展水平

此技术是否是当前可实现此目的的最先进的技术？这里，ISO/IEC/IEEE 29119和ISO 26262这类标准可以提供支持。ISO 26262标准甚至建议测试工程师根据ASIL等级使用适用的技术。要了解与ISO 26262标准中建议的偏差，请参阅第2.2章节关于ISO 26262的内容。

测试依据

测试依据是否为测试技术提供了合适的测试条件？例如，测试依据包含多个参数或变量，测试

中国软件测试认证委员会（CSTQB®）

工程师只能构建等价类。测试工程师必须能够将参数或变量的值分组到合理的等价类中。这同样适用于边界值的情况，只能在线性有序的取值范围内才能使用此技术测试。

基于风险的测试

基于风险的测试是指识别产品风险，并在选择技术时考虑风险级别。例如，仅当存在违反边界规则的可能，并由此会带来风险时，测试边界值才有意义。

测试级别

是否可以在相应测试级别合理使用该技术？如果在源代码或内部结构作为测试依据的情况下，则白盒测试特别适合。在理想情况下，结构的覆盖程度是可测量的。如果测试对象是可获取且可观测的，则黑盒测试比较适合。例如，在系统测试中测试传感器的等价类可能比在组件测试中更有效。如果某项测试设计技术在一个测试级别不适用，测试工程师应该根据测试策略选择其他替代测试级别或措施。

选择测试技术的示例

下表通过示例，介绍了前文提到的几方面因素是如何影响测试技术的选择的。

	测试设计技术	建议用于ASIL等级A?	测试依据是否适用?	缺陷未被检测到时，有风险?	“系统测试”级别是否合理?	选择
1	基于需求的测试	++	是	++	是	X
2	等价类划分	+	是	++	是	X
3	边界值分析	+	否	–	是	
4	语句测试	++	是	++	否	
5	判定测试	+	是	++	否	
6	MC/DC	+	是	+	否	
7	错误推测	+	否	++	是	
8	故障注入	+	是	+	否	
9	背靠背测试	+	否	++	是	

表6：选择测试技术的示例

中国软件测试认证委员会（CSTQB®）

附录

汽车数据库和通信协议

接口	数据库	通信协议
内存	ASAM MCD-2 MC（又称为ASAP2或A2L）	ASAM MCD-1 XCP（通用测量和标定协议） ASAM标准CCP（CAN标定协议）
总线	ASAM MCD2 NET 标准（又称为FIBEX-现场总线交换格式）	FlexRay（ISO 17458） CAN（基于ISO 11898-2的为控制器局域网）
	DBC（适用于CAN的通信数据库）	CAN（基于ISO 11898-2的为控制器局域网）
诊断	ASAM MCD2 D（又称为ODX）	KWP2000（ISO 14230） ISO-OBD（ISO 15031） UDS（ISO 14229）
	CDD（CANdelaStudio诊断描述）	

表7：汽车行业通用数据库和通信协议

AUTOSAR标准化了一种XML格式，该格式集成了整车的数据库。这就是ARXML格式（AUTOSAR应用程序接口主表，XML Scheme R3.0）。

ASAM代表“自动化及测量系统标准协会”。

表格清单

中国软件测试认证委员会（CSTQB®）

参考文献

[1] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, ISO/IEC TS 24748-1:2016 Systems and software engineering - Life cycle management - Part 1: Guide for life cycle management, 2016.

[2] Verband der Automobilindustriee.V. （VDA） / QMC Working Group 13 / Automotive SIG, Automotive SPICE Process Assessment Model, Berlin: Verband der Automobilindustrie e. V. （VDA）, 2008.

[3] AUTOSAR, http://www.autosar.org/specifications/, [Online].[Zugriff am 04 04 2016].

[4] ZVEI, Best Practice Guideline - Software Release, Frankfurt am Main: ZVEI,2016.

[5] International Software Testing Qualifications Board （ISTQB®） / German Testing Board e.V.（GTB）,ISTQB®/GTB Certified Tester Advanced Level （CTAL） Syllabus - Technical Test Analyst （TTA） - Deutsche Ausgabe, German Testing Board e.V. （GTB）, 2012.

[6] Verband der Automobilindustriee.V. （VDA） / QMC Working Group 13, Status and outlook VDA QMC working group 13 - Automotive SPICE 3.0, Blue-Gold Volume, in Sixth VDA Automotive SYS Conference, Berlin, 2016.

[7] Verband der Automobilindustriee.V. （VDA） / QMC Working Group 13 / Automotive SIG, Automotive SPICE Process Assessment / Reference Model, http://www.automotivespice.com/download/, 2015 Version 3.0.

[8] International Organization for Standardization （ISO）, ISO 26262:2011 Road Vehicles - Functional Safety, Genf, 2011.

[9] AUTOSAR, Glossary AUTOSAR Release 4.2.2, [Online].Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_TR_Glossary.pdf.[Zugriff am 03 03 2016].

[10] H. Wallentowitz, HandbuchKraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen; mitzahlreichenTabellen, Wiesbaden: Vieweg, 2016.

[11] K. Borgeest, Elektronik in der Fahrzeugtechnik, Springer Vieweg, 2014.

[12] MISRA Electrical Group MIRA Ltd., MISRA-C:2012-Programmierrichtlinien - Version 3, UK, Warwickshire, 2013.

中国软件测试认证委员会（CSTQB®）

[13] 754-2008 - IEEE Standard for Floating-Point Arithmetic, 754-2008 - IEEE Standard for Floating-Point Arithmetic, 2008.

[14] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, Institute of Electrical and Electronics Engineers （IEEE）, ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes, 2015-15-05.

[15] Verband der Automobilindustriee.V. （VDA）,

EntwicklungsoftwarebestimmterSysteme - Forderungen an Prozesse und Produkte, Bd. 13, Verband der Automobilindustriee.V. （VDA）, 2004.

[16] Measuring, Association for Standardization of Automation and,

http://asam.net/, 2016.[Online].[Zugriff am 2016].

[17] K. Hoermann, M. Mueller, L. Dittmann und J. Zimmer, Automotive SPICE in Practice in der Praxis-Interpretationshilfe für Anwender und Assessoren, Heidelberg: dpunktverlag GmbH, 2.Auflage, 2016.

[18] National Instruments Germany GmbH, Einsatz von Fault Insertion Units （FIUs）für die ÜberprüfungelektronischerSteuergeräte, Nr. 25.Juni, 2015.

[19] Patzer und Zaiser, Einsatzgebiete für XCP, in XCP-Das Standardprotokoll für die SteuergeräteEntwicklung, Stuttgart, Vector Informatik GmbH, 2014.

[20] International Software Testing Qualifications Board （ISTQB®） / German Testing Board e.V. （GTB）, ISTQB®/GTB Certified Tester Foundation Level （CTFL®） Syllabus - Version 2011 1.0.1 - Deutsche Ausgabe, German Testing Board e.V. （GTB）,2011.

[21] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, Institute of Electrical and Electronics Engineers （IEEE）, ISO/IEC/IEEE 29119-1:2013 Software and systems engineering - Software testing - Part 1: Concepts and definitions,2013-09-01.

[22] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes, International Organization for Standardization （ISO）,2008-02-01.

[23] A. Spillner, T. Roßner, M. Winter und T. Linz, Praxiswissen Softwaretest Testmanagement: Aus- und Weiterbildungzum Certified Tester - Advanced Level nach ISTQB®-Standard, Heidelberg: dpunkt.verlag, 2008.

[24] Verband der Automobilindustrie e.V. （VDA）, Sicherung der Qualität in der

中国软件测试认证委员会（CSTQB®）

Prozesslandschaft,Bd.Band 4, Verband der Automobilindustriee.V. （VDA）,2011.

[25] dpa,www.motor-talk.de, 24 02 2015.[Online].Available: http://www.motor-talk.de/news/die-zahl-der-modelle-waechst-der-absatz-nicht-t5219608.html.[Zugriff am 12 12 2016].

[26] AUTOSAR, Requirements on Acceptance Test AUTOSAR TC Release 1.1.0, [Online].Available:
http://www.autosar.org/fileadmin/files/standards/tests/tc-1-1/general_auxiliary/AUTOSAR_ATR_Requirements.pdf.[Zugriff am 2016 12 12].

[27] R. Schönfeld, Regelungen und Steuerungen in der Elektrotechnik, VerlagTechnik GmbH, 1993.

[28] AUTOSAR, Project Objectives AUTOSAR Release 4.2.1, [Online].Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_RS_ProjectObjectives.pdf.[Zugriff am 03 03 2016].

[29] AUTOSAR, Main Requirements AUTOSAR Release 4.2.1, [Online].Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_RS_Main.pdf.[Zugriff am 03 03 2016].

[30] G. Baumann, Was verstehenwirunter Test? Abstraktionsebenen, Begriffe und Definitionen FKFS 1.AutoTest; FachkonferenzzumThema Test und Diagnose in der Automobilentwicklung, Stuttgart, 2006.

[31] C. Hobbs, Embedded Software Development for Safety-Critical Systems, Taylor & Francis Group, 2016.

[32] AUTOSAR, AUTOSAR - The worldwide Automotive Standard for E/E systems, ATZ extra, p. 5, 2013.

[33] A. Spillner und T. Linz, Basiswissen Softwaretest [Elektronische Ressource]: Aus- und Weiterbildungzum Certified Tester - Foundation Level nach ISTQB®-Standard, Heidelberg: dpunkt.verlag, 2012.

[34] International Software Testing Qualifications Board （ISTQB®） / German Testing Board e.V. （GTB）, ISTQB®/GTB Standardglossar der Testbegriffe Version 3.1, Erlangen: German Testing Board e.V. （GTB）, 13.April 2016.

[35] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, Institute of Electrical and Electronics Engineers （IEEE）, ISO/IEC/IEEE 29119-3:2013 Software and systems engineering - Software testing - Part 3: Test documentation, 2013-09-01.

中国软件测试认证委员会（CSTQB®）

[36] AUTOSAR, Acceptance Test Main Requirements AUTOSAR TC Release 1.1.0, [Online].Available: http://www.autosar.org/fileadmin/files/releases/tc-1-1/general_auxiliary/AUTOSAR_ATR_Main.pdf.[Zugriff am 2016 03 03].

[37] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, Institute of Electrical and Electronics Engineers （IEEE）, ISO/IEC/IEEE 29119-4:2015 Software and systems engineering - Software testing - Part 4: Test techniques, Bd. 4, 2015.

[38] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, Institute of Electrical and Electronics Engineers （IEEE）, ISO/IEC/IEEE 29148:2011 - Systems and software engineering - Life cycle processes - Requirements engineering, 2011-12-01.

[39] M. Winter, M. Ekssir-Monfared, H. M. Sneed, R. Seidl und L. Borner, Der Integrationstest: Von Entwurf und ArchitekturzurKomponenten- und Systemintegration, München: Carl HanserVerlag GmbH & Co. KG, 2012.

[40] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, ISO/IEC 33020-03:2015 Informationstechnik - Prozessbewertung - Rahmenwerk für ProzessmessungenzurBeurteilung der Prozessfähigkeit, 01-03-2015.

[41] M. Conrad und G. Sandmann, A Verification and Validation Worflow for IEC 61508 Applications, SAE International, 2009.

[42] H.-W. Wiesbrock, M. Conrad, I. Fey und H. Pohlheim, EinneuesautomatisiertesAuswertverfahren für Regressions- und Back-to-Back-Tests eingebetteterRegelsysteme, Softwaretechnik-Trends, Bd. 22, 2002.

[43] U. Freund, V. Jaikamal und J. Löchner, Multilevel System Integration of Automotive ECUs based on AUTOSAR, [Online].Available: http://papers.sae.org/2009-01-0918/.[Zugriff am 27 09 2016].

[44] T. Ringler, C. Dziobek und F. Wohlgemuth, TagungsbandModellbasierteEntwicklungeingebetteterSysteme - Chancen und Herausforderungenbei der virtuellenAbsicherungverteilterBody&Comfort-Funktionen auf Basis von AUTOSAR - S.83 - 93, [Online].Available: https://www.in.tu-clausthal.de/fileadmin/homes/GI/Documents/MBEES15Proceedings.pdf.[Zugriff am 27 09 2016].

[45] Bakshi, U.A.; Baksi; V.U.: Control Systems, Edition 2010.（English edition）

[46] International Organization for Standardization （ISO）, International Electrotechnical Commission （IEC）, ISO/IEC 2382:2015-05 Information

中国软件测试认证委员会（CSTQB®）

technology - Vocabulary, 2015-05.

[47] International Organization for Standardization （ISO）, International Electrotechnical Commission（IEC）, ISO/IEC 24765:20170-09 Information technology - Vocabulary, 2017-09.

[48] German Association of the Automotive Industry （VDA） / QMC Working Group 13 / Automotive SIG, Automotive SPICE Process Assessment / Reference Model,http://www.automotivespice.com/download/, 2017, Version 3.1.

[49] International Software Testing Qualifications Board （ISTQB®） - ISTQB® Certified Tester Foundation Level （CTFL®） Syllabus - Version 2018.

中国软件测试认证委员会（CSTQB®）

定义

本文除了使用ISTQB®术语表[ISTQB® 2016]外，还使用了以下（非粗体）课程大纲专用术语。应按照以下定义使用这些术语：

术语	定义/含义	术语表关键术语[28]	参考
汽车开放系统架构（AUTOSAR） Automotive Open System Architecture (AUTOSAR)	2003年，全球相关行业各伙伴公司携手合作成立的一个国际联盟组织，其目标是为汽车行业创建一个开放、标准化的软件架构。		
汽车安全完整性等级 Automotive Safety Integrity Level	**用四个等级中的一个来指定ISO26262对项目或元素的必要需求和安全措施，以避免出现不合理的残余风险，其中“D”表示最高等级，需要最苛刻的安全需求，“A”表示最低等级，需要最一般的安全需求。**	X	[8]
Automotive SPICE	**适用于汽车工业的符合 ISO/IEC 33002:2015 要求的过程参考模型和相关过程评估模型。**	X	[9]
背靠背测试 back-to-back-testing	**一种测试类型，对测试项的两个或多个变体，或同一测试项的仿真模型，执行相同的测试用例并比较结果。 另请参阅“比较测试”。**	X	[32]
基础软件 Basic software	（AUTOSAR）：标准化的、以硬件为导向的软件组件。		[9]
断线测试盒 Breakout box	用于分析、中断或操控线束物理信号的测量单元。		[10]
总线系统 Bus system	多个电控单元为信息交互而连接形成的网络。		[11]
能力维度 Capability dimension	定义了按照能力级别划分的一系列过程属性。过程属性使得过程能力可度量。		[7]
能力指标 Capability indicator	可用于评估过程能力执行和情况说明的指标。		[7]
能力级别 Capability level	一个或多个充分满足的过程属性，可以显著提高过程能力。		[7]
闭环系统 closed-loop-System	**其控制或者输入会受输出或输出变化影响的系统。 另请参阅“开环系统”。**	X	[44]
代码评审 Code review	对照计划的目标，对代码进行适合性检查，并对照所提供的规格说明和标准，对代码进行偏离分析。		[7]
编码标准 coding standard	**一套标准，描述了数据或程序组件的设计特性或设计说明。**	X	[46]
组件HiL Components-HiL	用于测试单个ECU的HiL环境。		[10]
条件覆盖 Condition coverage	请参阅ISTQB®术语表3.1		

[28] GA版本发布后，粗体术语将被纳入到ISTQB®术语表中。

中国软件测试认证委员会（CSTQB®）

术语	定义/含义	术语表关键术语[28]	参考
条件测试 Condition testing	请参阅ISTQB®术语表3.1		
验证准则 Criteria for verification	验证软件所需的一组测试用例和准则。		[9]
网络安全（汽车） Cybersecurity (Automotive)	系统处于网络安全状态，以及为实现这一目标所采取的措施。		待定
指令（MISRA） Directive (MISRA)	MISRA-C:2012中未被静态分析工具完全验证的编程指南。		[12]
缺陷列表 Defect list	已修复和未修复的缺陷列表。通常是测试报告的一部分。		[4]
E/E系统 E/E-System	电气元件或电子元件组成的功能系统。		[8]
ECU抽取 ECU extract	从系统配置描述中抽取出的单个电控单元数据。		[9]
ECU配置描述 ECU configuration description	包含用于在电控单元上集成软件组件的数据		[9]
环境模型（汽车） environment model (Automotive)	**在实时仿真中，对组件或系统所在的真实环境（包括其它组件、车辆、环境）的抽象。**	X	**[10]**
电气故障仿真 Electrical Error Simulation	请参阅“故障注入单元”		
定点 Fixed point	由固定位数组成的数字。小数点的位置是固定的。		
浮点 Floating point	实数的一种近似表示法。		[13]
故障注入 Fault injection	请参阅ISTQB®术语表3.1		
故障注入单元 Fault insertion unit	测试环境的一部分，能够仿真组件或系统接口上的缺陷。		
功能安全 functional safety	**没有因电气/电子（E/E）系统失效行为造成的危害而带来的不合理风险。**	X	**[3]**
硬件在环 Hardware in the Loop	**在仿真环境中，使用包含软件的真实硬件进行的动态测试。**	X	**[4]**
安装建议 Installation recommendation	会被添加到软件发布中。供应商使用这些信息向OEM确认发布项在用于公共道路时，可不受限制，而且可以在公共道路上使用/测试。		
功能列表 List of functions	在某一个软件版本实现的功能需要在版本规划阶段定义，且在功能列表中表述。		[4]

国际软件测试认证委员会（CSTQB®）

术语	定义/含义	术语表关键术语[28]	参考
渗透/浸泡测试 soak test	浸泡测试类似于根据现场经验进行的测试，但使用较大样本的普通用户作为测试工程师，并且不局限于先前指定的测试场景，而是在日常生活中的真实条件下执行测试。如果需要确保测试工程师的安全，可以对这些测试进行限制，例如，使用更多安全措施或禁用执行器。		[8]
方法表（汽车） method table (Automotive)	**包含不同测试方法、测试技术和测试类型的表格，根据汽车安全完整性等级（ASIL）和测试对象的环境来选择使用表中的内容。**	X	[8]
模型在环 Model in the Loop	**在模拟环境中，使用系统的仿真模型进行的动态测试。**	X	[4]
修订条件/判定覆盖测试（MC/DC 覆盖测试） Modified condition/decision testing (MC/DC-Test)	请参阅ISTQB®术语表3.1。		
条件组合测试 Multiple condition testing	请参阅ISTQB®术语表3.1		
开环系统 open - loop-system	**一类系统，在这个系统中控制动作或输入不受输出或输出变化的影响。 另请参阅“闭环系统”。**	X	[44]
原厂设备制造商（OEM） Original equipment manufacturer (OEM)	在汽车行业中，这个术语用于表示汽车制造商。另请参阅“一级供应商……N 级供应商”		[2]
产品开发过程 Product development process	包括从最初的产品构思开始，直到生产开始在内的所有活动的过程。		[15]
生产 Production	生产已开发完成的产品。 在汽车环境中的PEP中，又称为制造/量产。		[14, 1]
过程属性 Process attribute	为了过程能力评估而存在的过程可度量特性。		[7]
过程维度 Process dimension	对所有过程按照类别进行了划分，类别下（第二级分类）又有不同的组。		[7]
过程改进 Process improvement	请参阅ISTQB®术语表3.1		
过程模型 Process model	请参阅ISTQB®术语表3.1		
产品生命周期 Product lifecycle	请参阅“系统生命周期”		
发布说明 Release	关于发布项的已实施功能、属性和既定用途的声明。[15]		[15]
发布项 Release item	具有规定的功能、属性和用途的可明确识别的元素。[15]		[6]
发布过程 Release process	发布产品的过程。		[4]

中国软件测试认证委员会（CSTQB®）

术语	定义/含义	术语表关键术语[28]	参考
发布目的 Release purpose	发布项可以实现或可能实现的目的。		[4]
发布建议 Release recommendation	测试工程师或测试经理根据测试结果提出的关于是否发布的建议		[4]
实时 Real time	在计算机系统的运行过程中，用于处理数据的程序时刻准备就绪，以便能够在预定的时段内得到处理结果。根据应用程序的不同，可以在临时随机分布的时间内或在预定的时间内生成数据。		[45]
具备实时处理能力的计算机 Real time capable computer	一种计算单元，能够保证在指定的时间段内处理好信号。		[10]
参考过程 Reference Process	请参阅ISTQB®术语表3.1		
回归测试策略 Regression test strategy	回归测试策略定义了以下内容：如果测试项发生变化，在选择回归测试用例时，应遵循哪些准则。		
剩余总线仿真 Rest bus simulation	对不存在的电控单元的总线接口进行虚拟化。		
规则（MISRA） Rule (MISRA)	MISRA-C:2012中可通过静态分析工具进行验证的编程指南。		[12]
运行时环境（AUTOSAR） Runtime environment (AUTOSAR)	用于控制和实施以下内容的抽象层：AUTOSAR 软件组件之间的数据交换，应用程序和基础软件（BSW）之间的数据交换，以及控制单元内外的数据交换。		[9]
安全文化 Safety Culture	整个公司共同开发功能安全产品时所表现出的态度。		[8]
安全生命周期 Safety lifecycle	安全相关系统的产品生命周期。从产品构思开始，直至废弃产品为止。		[8]
仿真时间 Simulation time	计算机仿真的时间。		[10]
软件组件 Software component	（AUTOSAR）：独立于硬件的软件层，包括单个应用程序和功能。		[9]
软件在环（SiL） Software in the loop (SiL)	**在仿真环境中，使用真实软件或试验性硬件的动态测试。**	X	**[4]**
软件合格性测试（ASPICE） Software qualification test (ASPICE)	**对完整的、集成好的软件进行的测试，以证明该软件是否符合软件需求。**	X	**[9]**
系统HiL System-HiL	一种用于仿真从电控单元组，到整个车辆的测试环境。		[10]

中国软件测试认证委员会（CSTQB®）

术语	定义/含义	术语表关键术语[28]	参考
系统集成测试（ASPICE） System integration test (ASPICE)	对系统架构设计进行测试，以证明集成系统是否符合系统架构设计，包括系统之间的接口。		[9]
系统配置描述 System configuration description	集成一辆车中所有电控单元时用到的数据。		[9]
系统生命周期 System lifecycle	系统退役之前，除PEP阶段之外的开发和实施阶段。		[15,1,14]
系统合格性测试（ASPICE） System qualification test (ASPICE)	**对完整的、集成好的系统（包括软件组件、硬件组件和机械部件）进行测试，以证明它们是否满足系统需求，以及整个系统是否可供交付。**	X	**[9]**
综合系统测试 System of systems testing	对综合系统进行测试，以验证是否满足指定的需求。		
测试项 Test item	1. 请参阅ISTQB®术语表 2. 汽车环境中的测试项由软件配置组成，包括基本的参数化，硬件和机械。[6]		[4]
1级供应商……n级供应商 Tier 1…n	在供应链中，不同级别的供应商依次称为1级供应商……n级供应商。OEM的直接供应商称为1级供应商，1级供应商的供应商称为2级供应商，依此类推。		[2]
测试文档 Test documentation	描述系统或组件测试计划或结果的文档。[ISO/IEC/IEEE 24765]		[46]
测试策略 Test strategy	请参阅ISTQB®术语表3.1		
可追溯性 Traceability	请参阅ISTQB®术语表3.1		
验证准则 Verification criteria	验证准则定义了成功验证测试项必须满足的定性和定量标准。		[7]
验证策略 Verification strategy	用于验证测试项的高级别计划，包括验证标准、验证活动及相关方法、技术和工具，以及待验证的工作产品或过程。		[7]
XiL测试环境 XiL test environment	**在不同虚拟测试环境中进行的动态测试的通用术语。 另请参阅硬件在环、软件在环、模型在环。**	X	

中国软件测试认证委员会（CSTQB®）

缩写

本课程大纲中使用了以下缩写：

缩写	缩写英文全称	定义/含义	参考
ACQ	Acquisition	采集	[7]
ASIL	Automotive Safety Integrity Level	汽车安全完整性等级	[8]
ASAM	Association for Standardisation of Automation and Measuring Systems	自动化及测量系统标准协会	[18]
ASPICE	Automotive SPICE	Automotive SPICE	
AUTOSAR	Automotive Open System Architecture	汽车开放系统架构	[9]
AUTOSIG	Automotive Specific Interest Group	汽车特殊利益集团	[17]
BP	Base Practice	基本实践	[7]
BSW	Base Software	基础软件	[9]
CTFL®	Certified Tester Foundation Level	基础级认证测试工程师	
E/E	Electric / Electronic	电气/电子	
ECU	Electronic Control Unit	电控单元	
EES	Electrical Error Simulation	电气故障仿真	[16]
EOP	End-of-Production	生产结束	
FIU	Fault Insertion Unit	故障注入单元	[18]
GP	Generic Practice	通用实践	[7]
HiL	Hardware-in-the-Loop	硬件在环	
IEC	International Electrotechnical Commission	国际电工委员会	
ISO	International Organization for Standardization	国际标准化组织	
ISTQB®	International Software Testing Qualifications Board	国际软件测试资质认证委员会	
MAN	Management (ASPICE)	管理（ASPICE）	[7]
MC/DC	Modified Condition/Decision Coverage	MC/DC 覆盖	

中国软件测试认证委员会（CSTQB®）

缩写	缩写英文全称	定义/含义	参考
MIL	Model in the loop	模型在环	
MISRA	Motor Industry Software Reliability Association	汽车工业软件可靠性协会	
OEM	Original Equipment Manufacturer	原厂设备制造商	
PA	Process Attribute	过程属性	[7]
PEP	Product Evolution Process	产品进化过程	[15]
PIM	Process Improvement (ASPICE)	过程改进（ASPICE）	[7]
QM	Quality Management	质量管理	
REU	Reuse (ASPICE)	复用（ASPICE）	[7]
RTE	Run Time Environment	运行时环境	[9]
SIL	Software in the Loop	软件在环	
SOP	Start-of-Production	开始生产	
SPICE	Software Process Improvement and Capability Determination	软件过程改进和能力测定	[7]
SPL	Supply (ASPICE)	供应（ASPICE）	[7]
SUP	Support (ASPICE)	支持（ASPICE）	[7]
SW	Software	软件	
SW-C	Software Component	软件组件	[9]
SWE	Software Engineering (ASPICE)	软件工程（ASPICE）	[7]
SYS	System Engineering (ASPICE)	系统工程（ASPICE）	[7]
VDA	German Association of the Automotive Industry	德国汽车工业协会	
WP	Work Product	工作产品	[7]
XCP	Universal Measurement and Calibration Protocol	通用测量和标定协议	[19]
XIL	Stands as upper tem for different in the Loop	各种在环测试技术的总称	

中国软件测试认证委员会（CSTQB®）