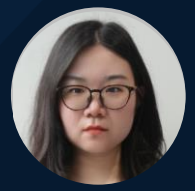
19th Asian Games Hangzhou 2022

GEELY 吉利控股集团

DevOps赋能软件开发质量提升

林燕，吉利汽车研究院（宁波）有限公司



2024 MathWorks
中国汽车年会

汽车行业现状

安全与合规性的重视

汽车制造商需要有效保证产品的安全性与可靠性，提高用户的信任和满意度;

软件定义汽车的崛起

软件不再是依附在硬件上的固定代码，而是连接最终用户的桥梁与纽带，软件的目的不仅仅是实现功能，更重要的是服务客户，生命周期和边界形态不断进化;

“新四化”及AI趋势

汽车智能化、网联化、共享化和电动化的趋势推动车载软件市场迅速增长，软件与高性能计算硬件的结合为汽车行业注入新的活力;

消费者对新功能的快速体验需求

消费者对功能体验要求更高，想要快速体验新功能的需求;

存在的痛点

车载软件架构复杂性

随着功能增加，车载软件系统变得极其复杂，要求高度的系统和模块集成，也因此容易出问题影响系统稳定性。

快速迭代带来的巨大压力

时间不足、匆忙发布、测试不足，造成质量问题频发

激烈的市场竞争

大量新势力的涌入导致竞争加剧


“卷”
成为行业现状的必然结果

从软件的角度看待汽车行业的“卷”

存在的痛点

- **车载软件架构复杂性**
  随着功能增加，车载软件系统变得极其复杂，要求高度的系统和模块集成，也因此容易出问题影响系统稳定性。

- **快速迭代带来的巨大压力**
  时间不足、匆忙发布、测试不足，造成质量问题频发

- **激烈的市场竞争**
  大量新势力的涌入导致竞争加剧

汽车开发面临的挑战

造车周期从36个月缩短至18个月甚至更短，软件要快速迭代

软件复杂度呈指数级增加

软件质量要求更高

单车软件开发成本降低

对软件团队的挑战

- 软件开发周期缩短1/2
- 功能需求很多且在不断迭代
- 软件是多车型并行交付的
- 如何做好软件管理
- 如何保证软件质量
- 如何提高人效

举例

工作量 “大”

- 工作量大
  - 环境配置内容多
  - 配置工作是重复性的
  - 编译构建步骤繁琐
- 配置时间短
  - 软件2~3周发布一次，集成配置最多1周时间
  - 集成测试内容多，手动测试繁琐
- 工具限制
  - 不支持多人同时配置
  - 有些配置错误无法提前发现
  - 单个接口配置时间长

管理过程 “难”

项目经理

软件释放风险不可控：

- 临时加班追赶或压缩测试团队时间？
- 软件总是延期释放？影响太大了

集成工程师

- 集成工程师认为他们的工作是重复性且无趣的，像机器人

项目交付 “不可控”

- 加班追赶或压缩测试团队的时间，软件质量无法保障。
- 延期释放，造成整个软件发布存在风险，且不可控；
- 团队目标不一致，无集体成就感和责任感

有没有办法可以兼顾“快速交付”和“质量”？

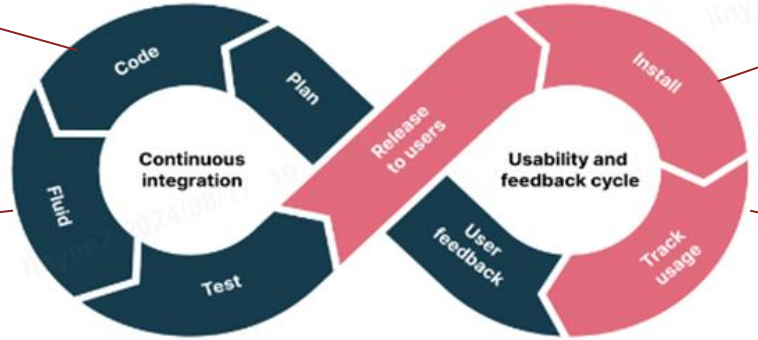
DevOps “快速交付” “质量保障”

自动化

自动化脚本和工具的使用，减少手动操作，提高开发、测试和部署的效率，降低人为错误的风险。

监控与反馈

持续监控，使团队能够及时发现问题并做出改进。

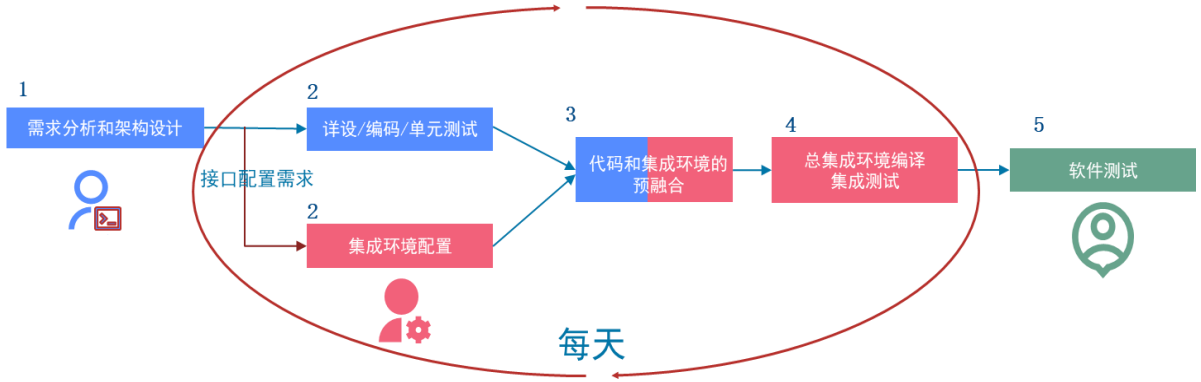


持续集成与持续交付

通过自动化测试和部署流程，频繁地将代码集成到主干，并快速交付到生产环境，以减少集成问题和加速交付节奏。

团队协作

强调跨团队合作，打破传统的职能壁垒，增强信息交流，以实现更高效的工作流程。



工具

集成配置自动化、
代码拉取自动化、
集成构建自动化、
集成测试自动化、

方法规范

接口配置标准化定义、
代码管理标准化、

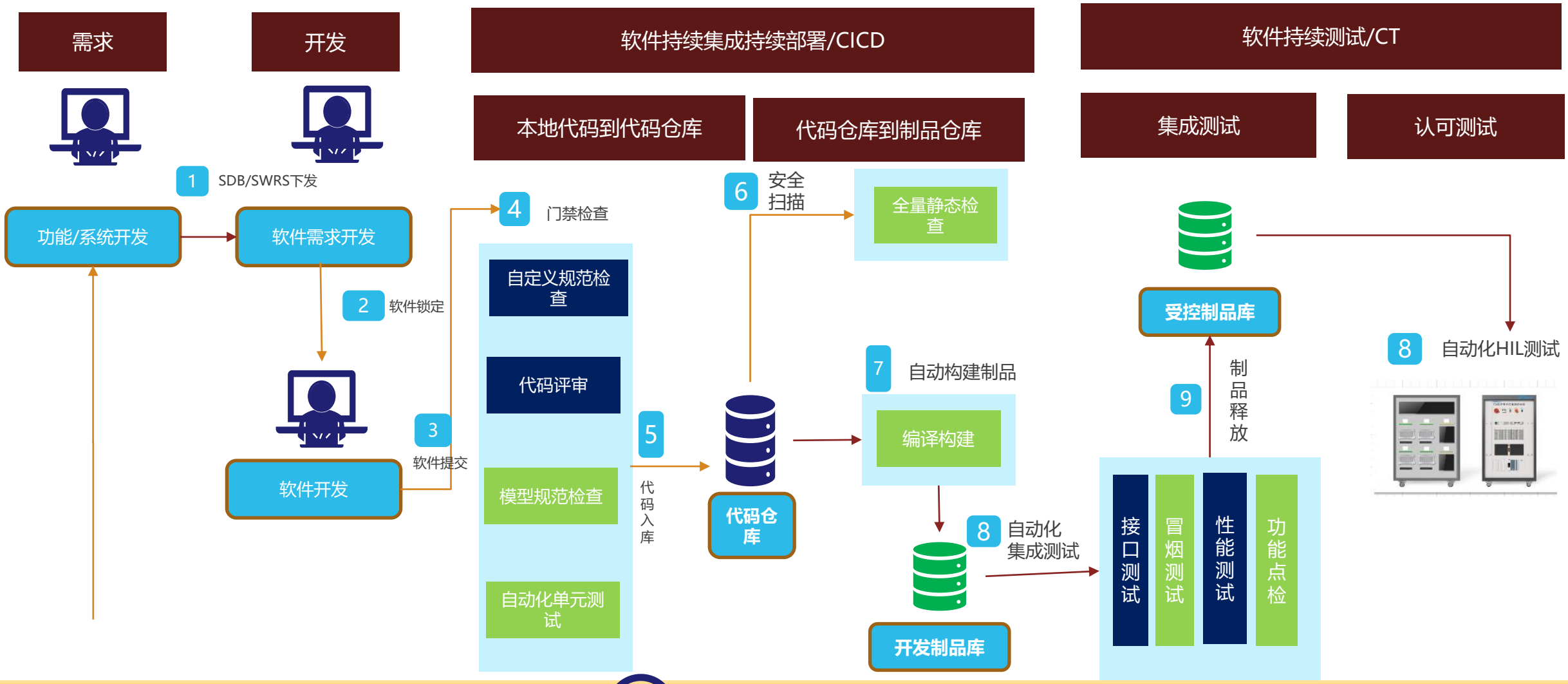
每日构建：提前发现编译问题
每日测试：提前发现软件问题

达成效果

- 集成配置和构建周期缩短到1~2天，缩短反馈环，提升响应周期，提前发现问题；
- 每日构建和每日测试，提前发现软件编译和软件问题，提升工作效率；
- 集成一次性通过率 > 90%，做好软件准出到测试的守门员；

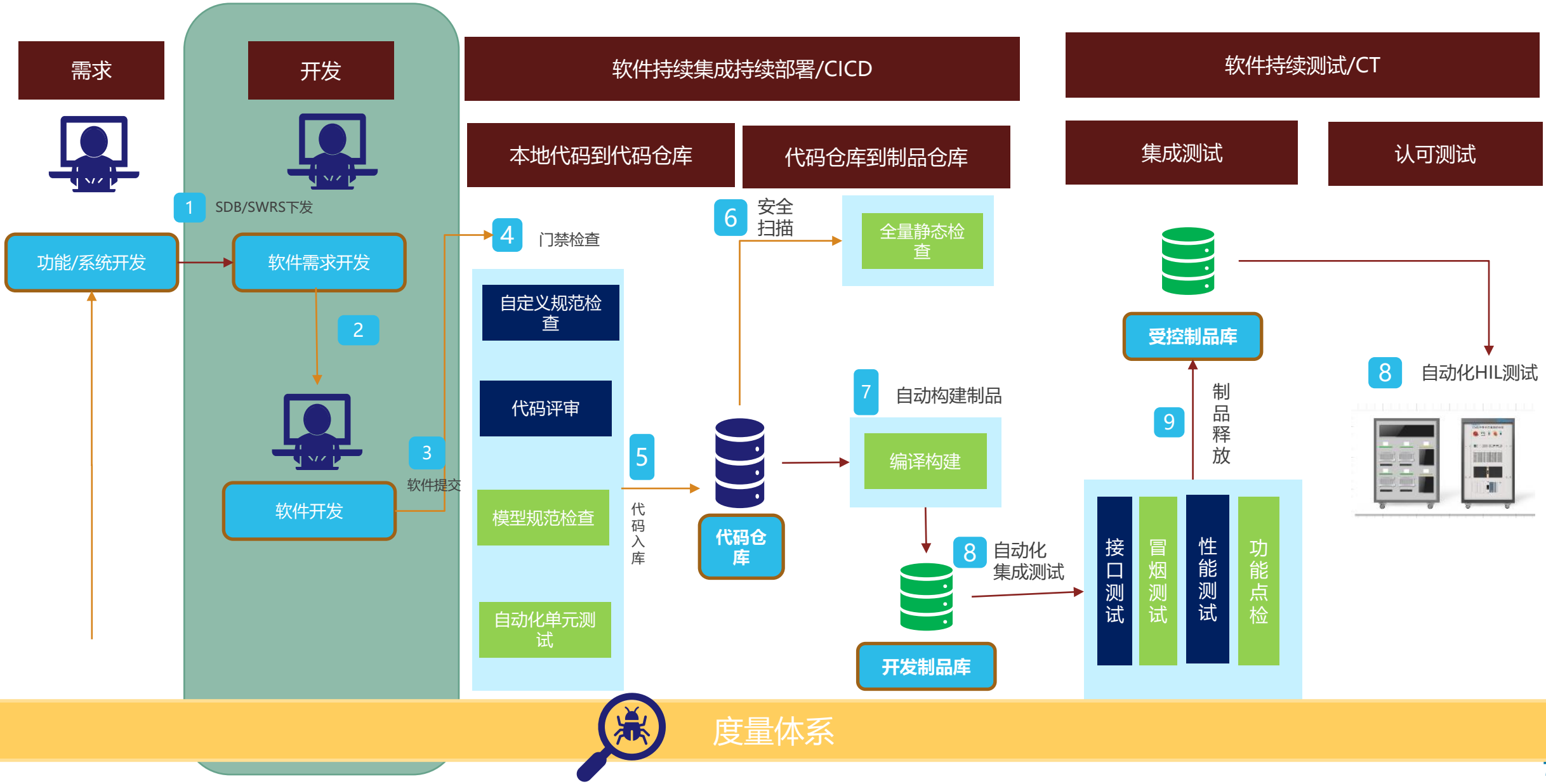
1、DevOps实践

➢ 以区域控制器为例，DevOps在软件开发全过程的实践路径。



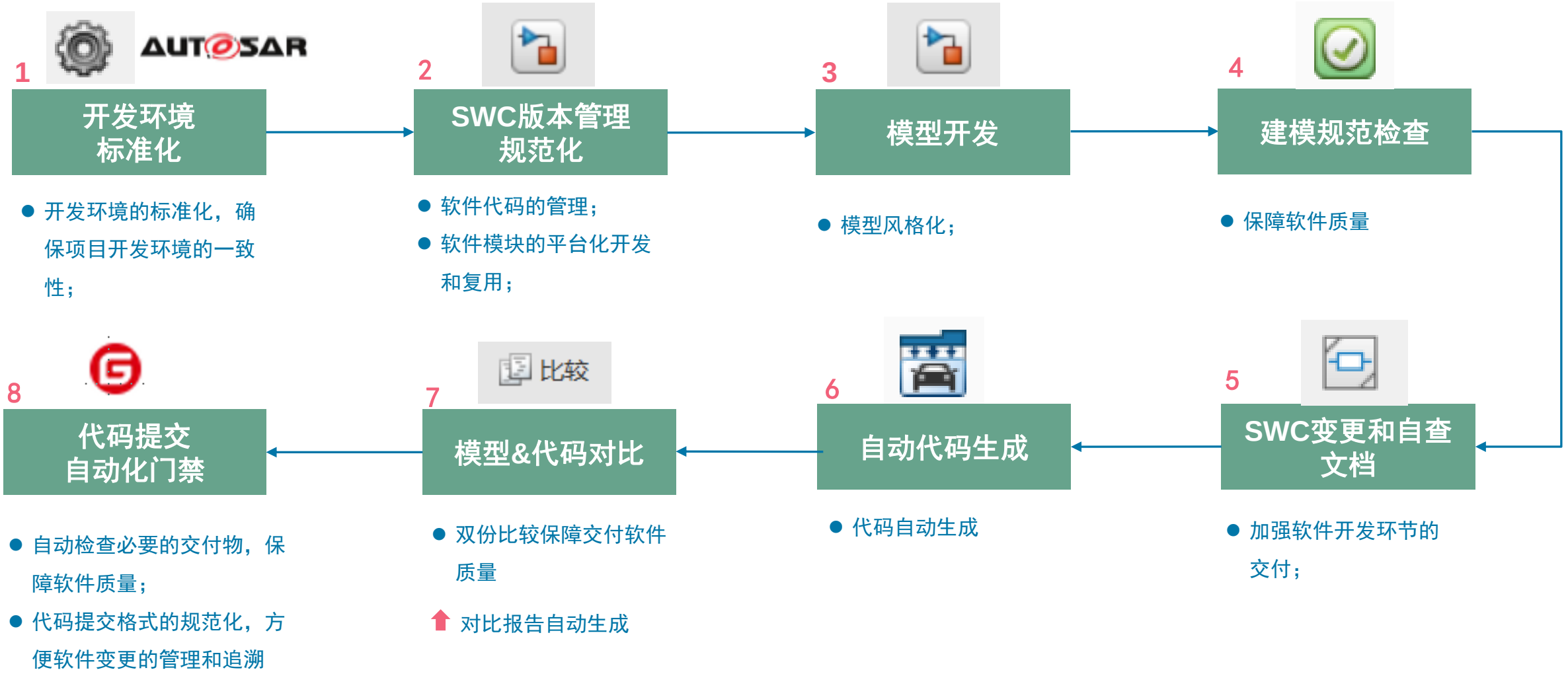
度量体系

2、DevOps在开发阶段的实践

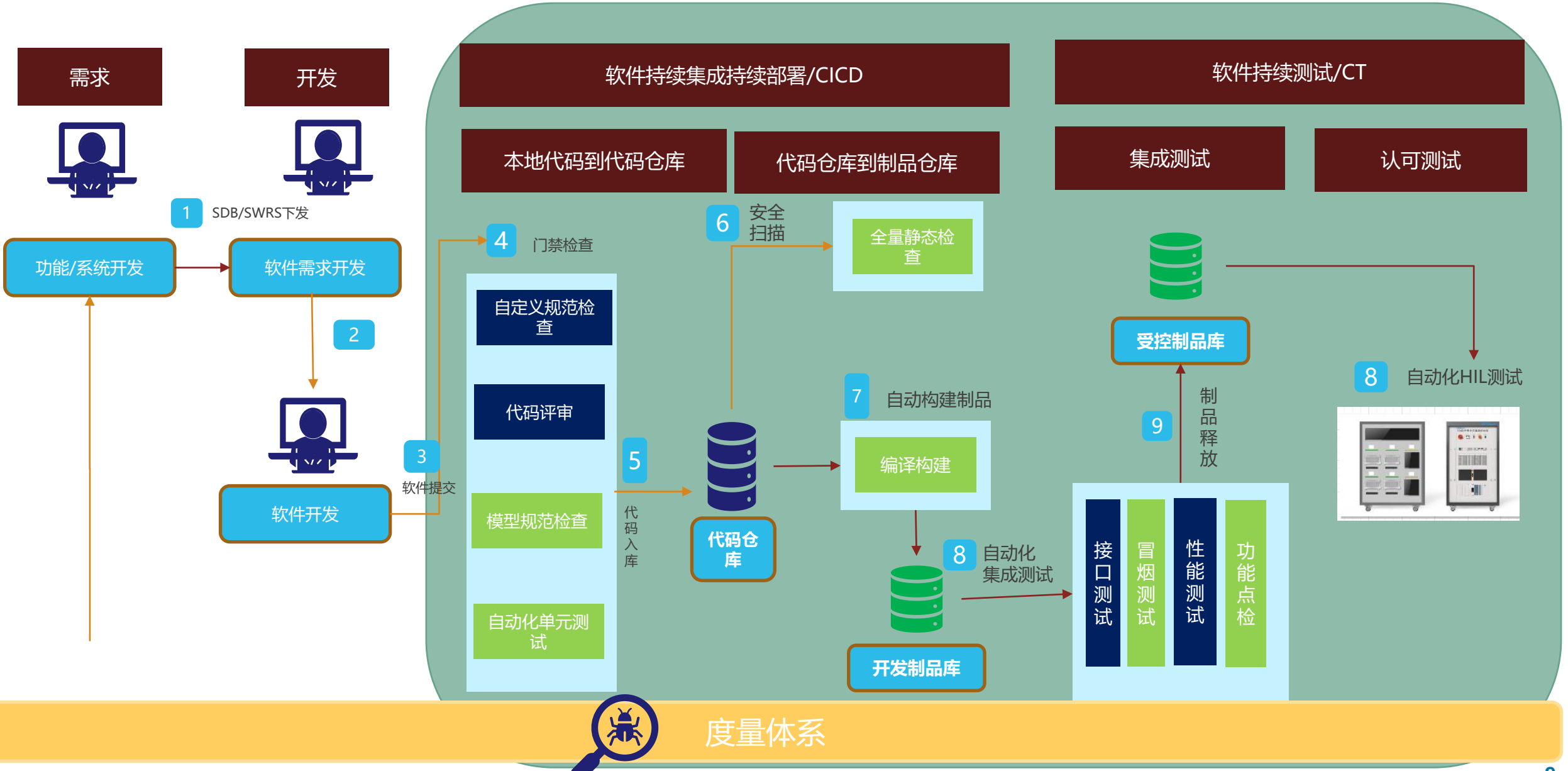


2.1 Simulink助力软件开发环节的自动化和过程质量保障

在软件开发环节，从工具和流程上，制定完整的开发规则，定义清晰的交付物清单，通过自动化的工具进行检查。



3、DevOps在CICT的实践

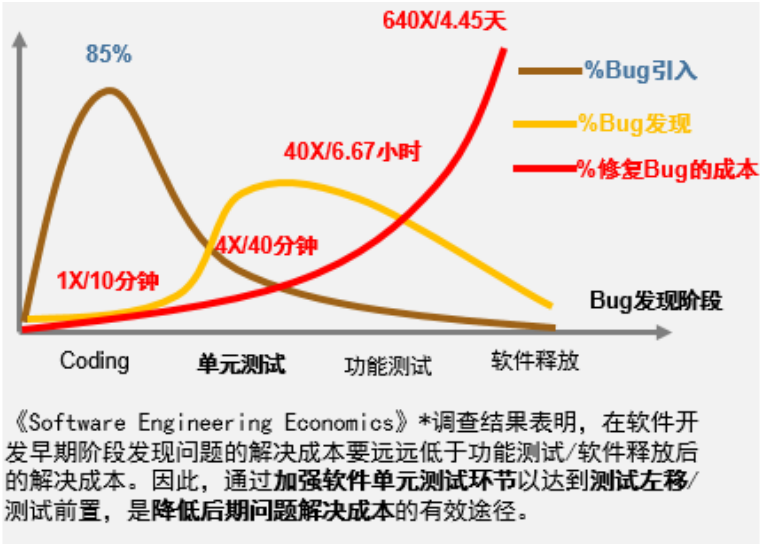


度量体系

Simulink助力CICT持续集成持续测试，构建软件质量的防护网

- 软件开发V模型右侧，全部实现自动化测试，提高软件开发效率；
- CICT环节，实现自动化流转和反馈机制，持续迭代和反馈；
- 不同维度的测试分层分级机制，实现交付周期和软件质量之间的平衡；

测试左移的价值驱动



《Software Engineering Economics》*调查结果表明，在软件开发早期阶段发现问题的解决成本要远远低于功能测试/软件释放后的解决成本。因此，通过**加强软件单元测试环节**以达到**测试左移/测试前置**，是**降低后期问题解决成本**的有效途径。

3.1 UnitCT定制化价值

原有工具的限制

- 基于Simulink Test二次开发的单元测试工具
- 简单好用
- 图形化
- 已经开发了许多测试用例

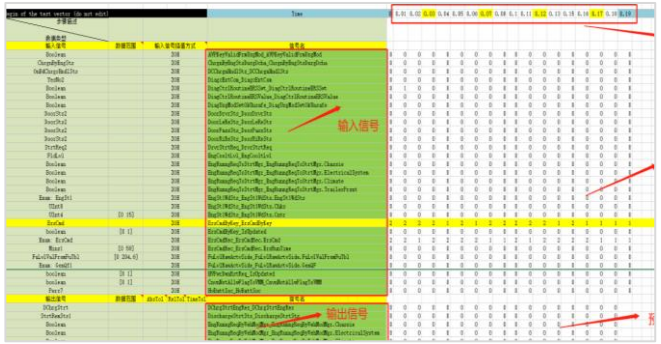
- 模型接口变更后，需要重新生成Test_Harness，测试用例无法直接复用
- 单元测试运行时间长，占用开发工程师的电脑，期间无法进行开发工作，不够高效
- 模型数据复杂时偶发Bug

深度合作：定制化开发单元测试工具UnitCT

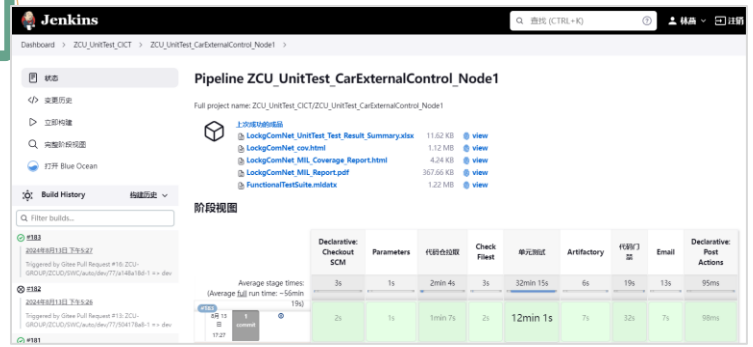
UnitCT的效果

1）可视化界面

- 支持时间序列和测试序列形式的测试



2）支持CI部署：服务器进行自动化单元测试



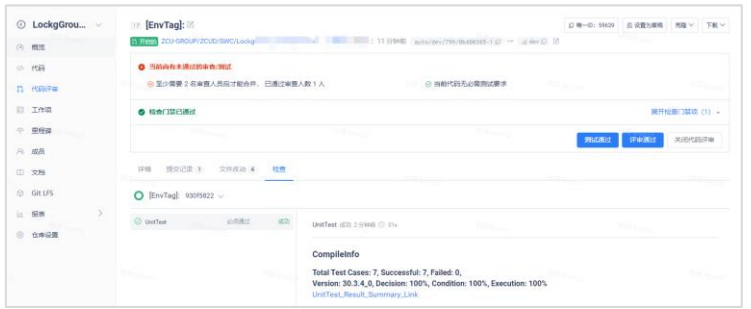
定制化单元测试工具

3）原有用例的一键式转换

- 原有测试用例仍可复用，减少因工具切换造成的无效人力投入

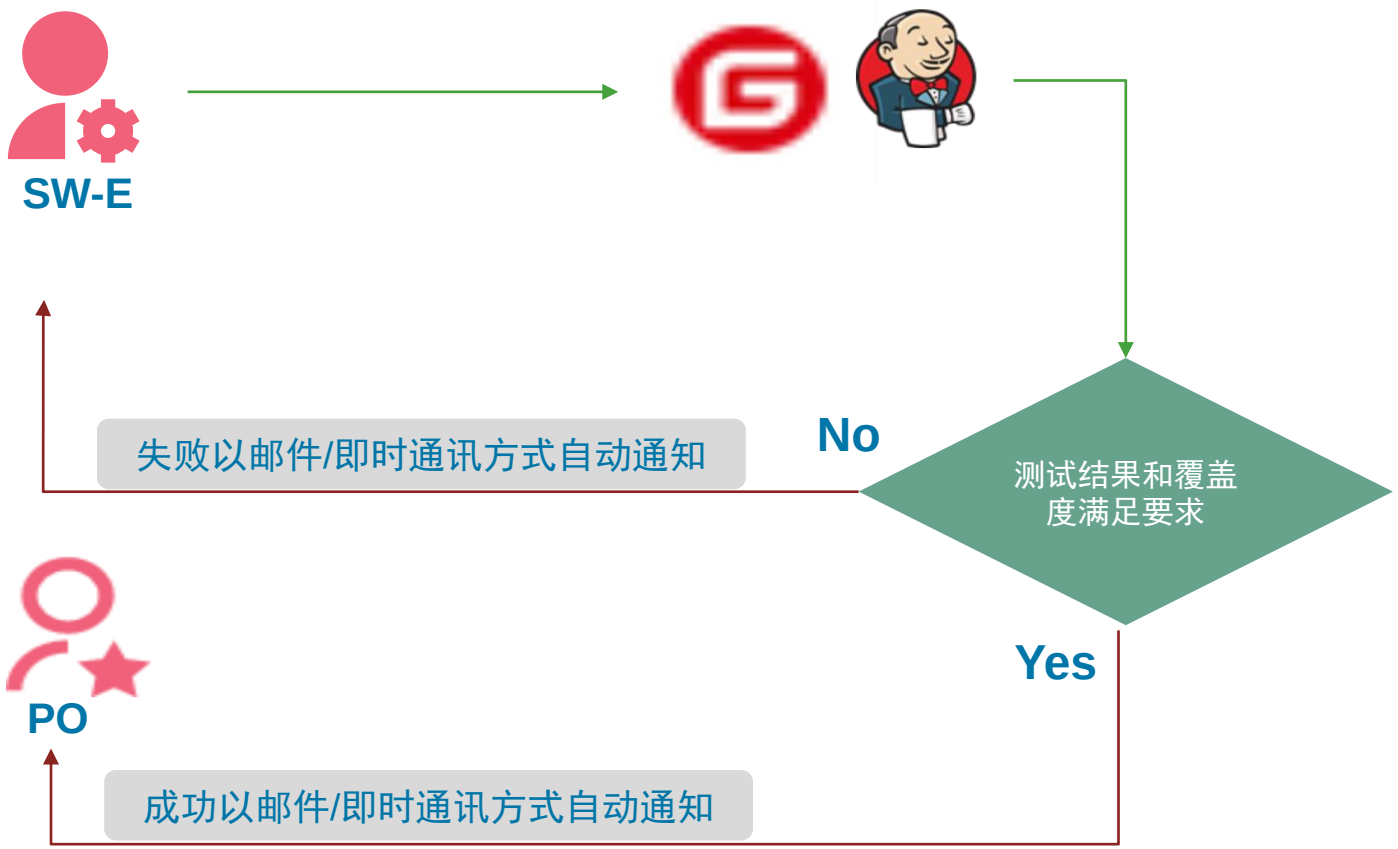
4）自动化代码门禁检查

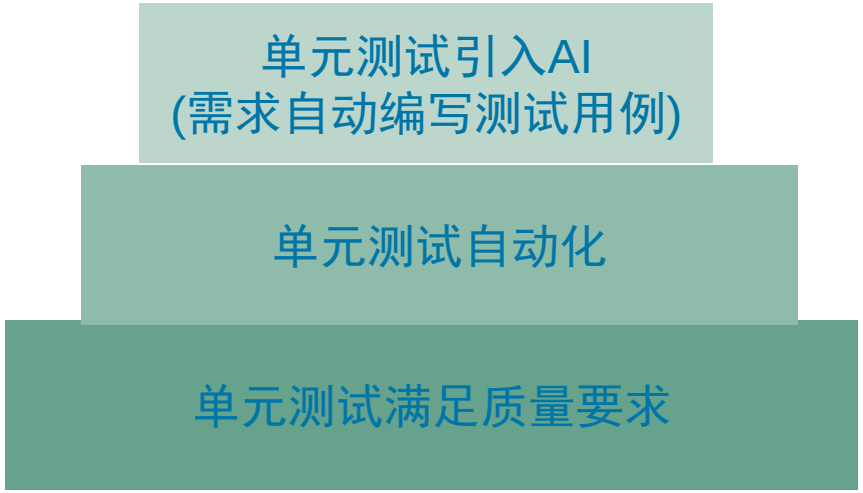
- 通过质量门禁，严格准入准出条件，提前暴漏问题和质量



自动化单元测试的总结

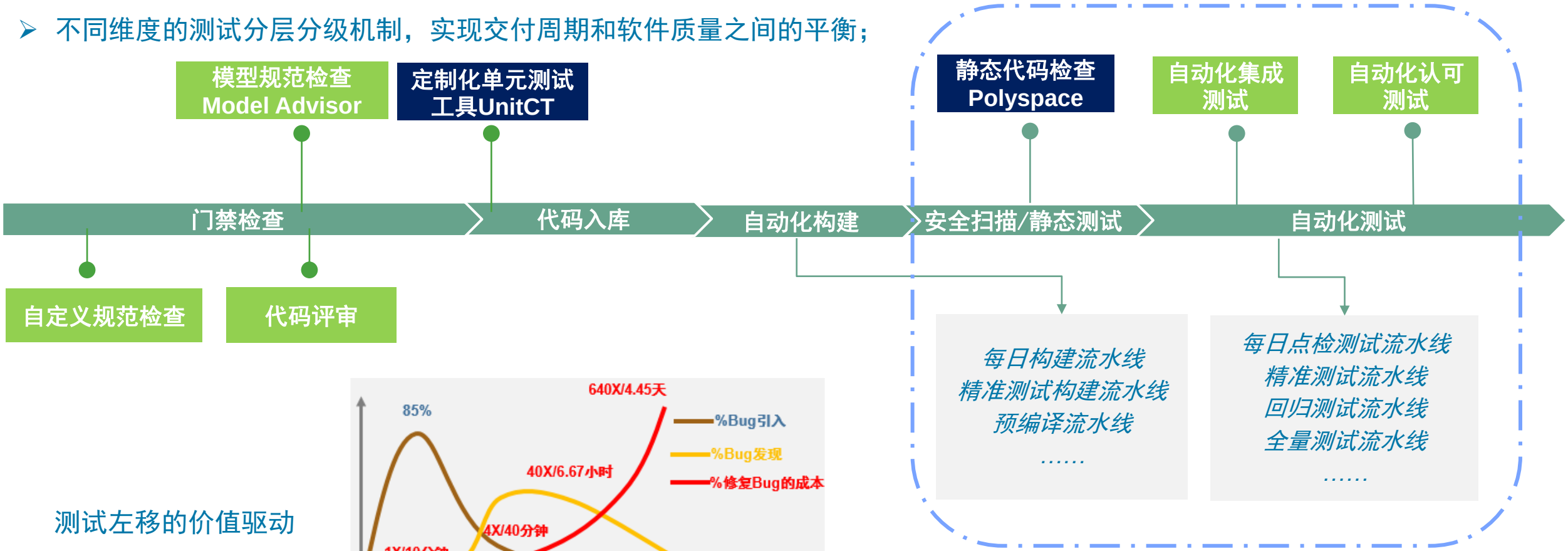
UnitCT定制化单元测试工具实现了单元测试的自动化和线上化，显著提升了工程师的软件开发效率,自动化成为我们软件质量管理的重要助手。





Simulink助力CICT持续集成持续测试，构建软件质量的防护网

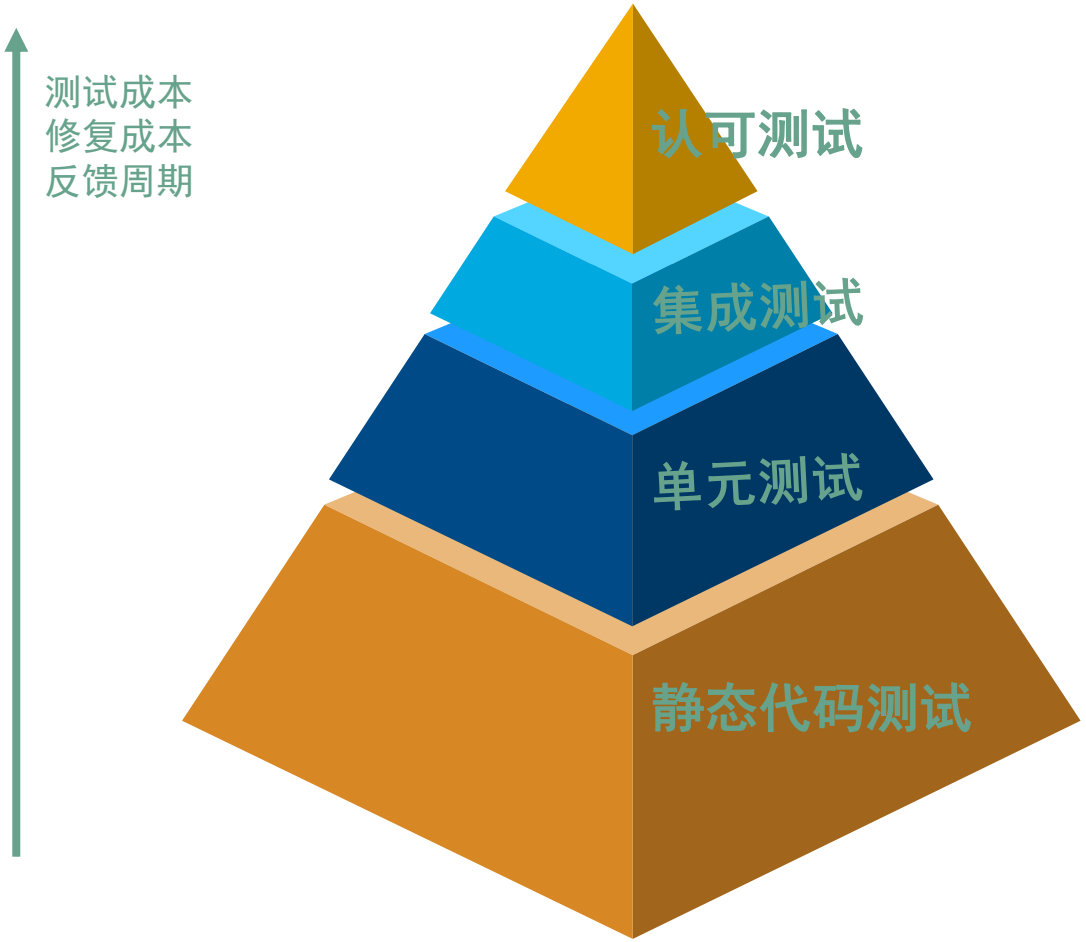
- 软件开发V模型右侧，全部实现自动化测试，提高软件开发效率；
- CICT环节，实现自动化流转和反馈机制，持续迭代和反馈；
- 不同维度的测试分层分级机制，实现交付周期和软件质量之间的平衡；



测试左移的价值驱动

《Software Engineering Economics》*调查结果表明，在软件开发早期阶段发现问题的解决成本要远远低于功能测试/软件释放后的解决成本。因此，通过**加强软件单元测试环节**以达到**测试左移/测试前置**，是**降低后期问题解决成本**的有效途径。

3.2 静态代码测试的价值



提高代码质量

发现潜在错误和不符合编码规范的问题；

提高开发效率

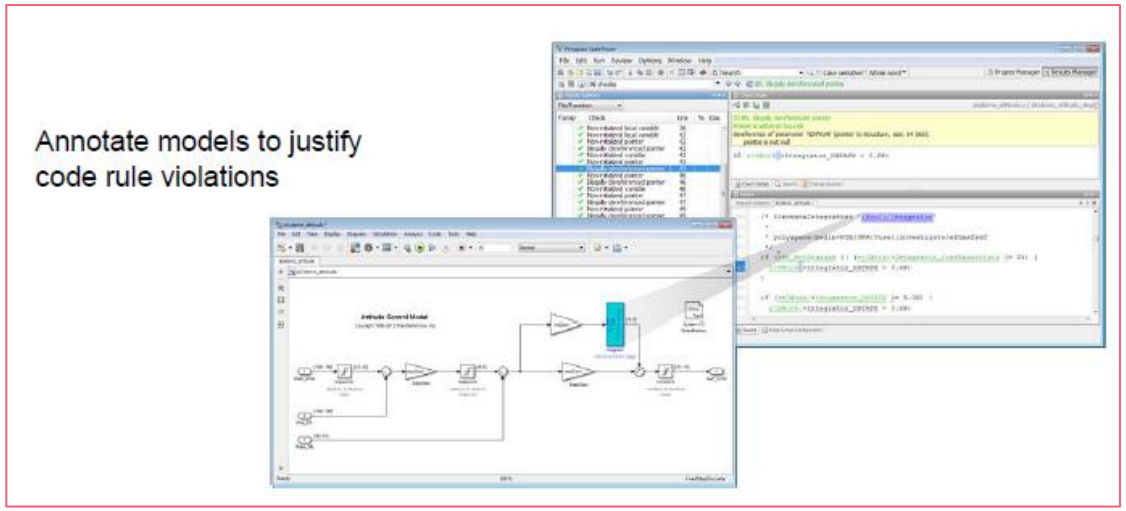
快速定位问题，减少回归测试时间；

节省成本

避免在动态测试中修复更复杂的问题；

静态代码测试工具-Polyspace

易用性和实用性是工具成功的基础。MBD建模开发工程师更注重功能逻辑的实现，往往在源码阅读能力上较弱。由于模型生成的代码可读性较差，因此对静态代码检查工具的可视化期望更高。

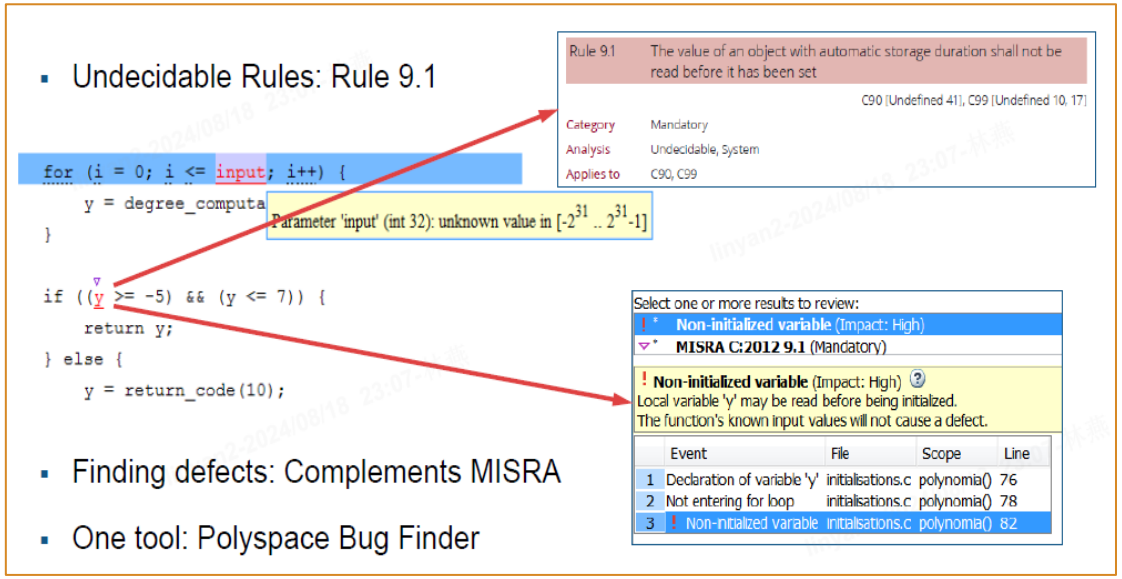


Polyspace在这方面表现出色：

- 能够从代码分析追溯到模型设计
- 提供完整的代码规范问题分析及解决方案建议
- 构建静态代码检查与建模规范的生态系统

通过这些优势，Polyspace为工程师提供了更高效的开发支持。

但是，Polyspace更强大的地方在于深度语义分析法：通过对代码的深层理解和解析来发现潜在的缺陷和验证程序的正确性。这种分析超越了传统的静态分析，利用形式化方法分析程序的语义特征。



4、DevOps实践总结

➢ 在区域控制器的开发过程中，DevOps的实施为团队从效率提升、资源优化、质量提升带来了显著的效益；

效率提升

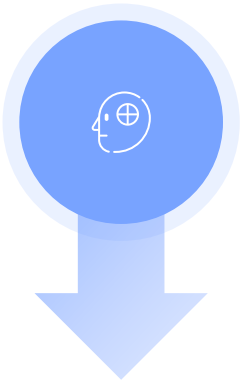
缩短开发周期：
自动化和标准化流程，减少了手动操作时间；

快速迭代：
团队能够更快地进行迭代与反馈，适应车型项目和需求变化的能力提升；

资源优化

提高资源利用率：
自动化后，团队可以将更多精力集中在高价值的任务上；

降低成本：
减少了人力投入和项目延误，降低了整体开发成本；

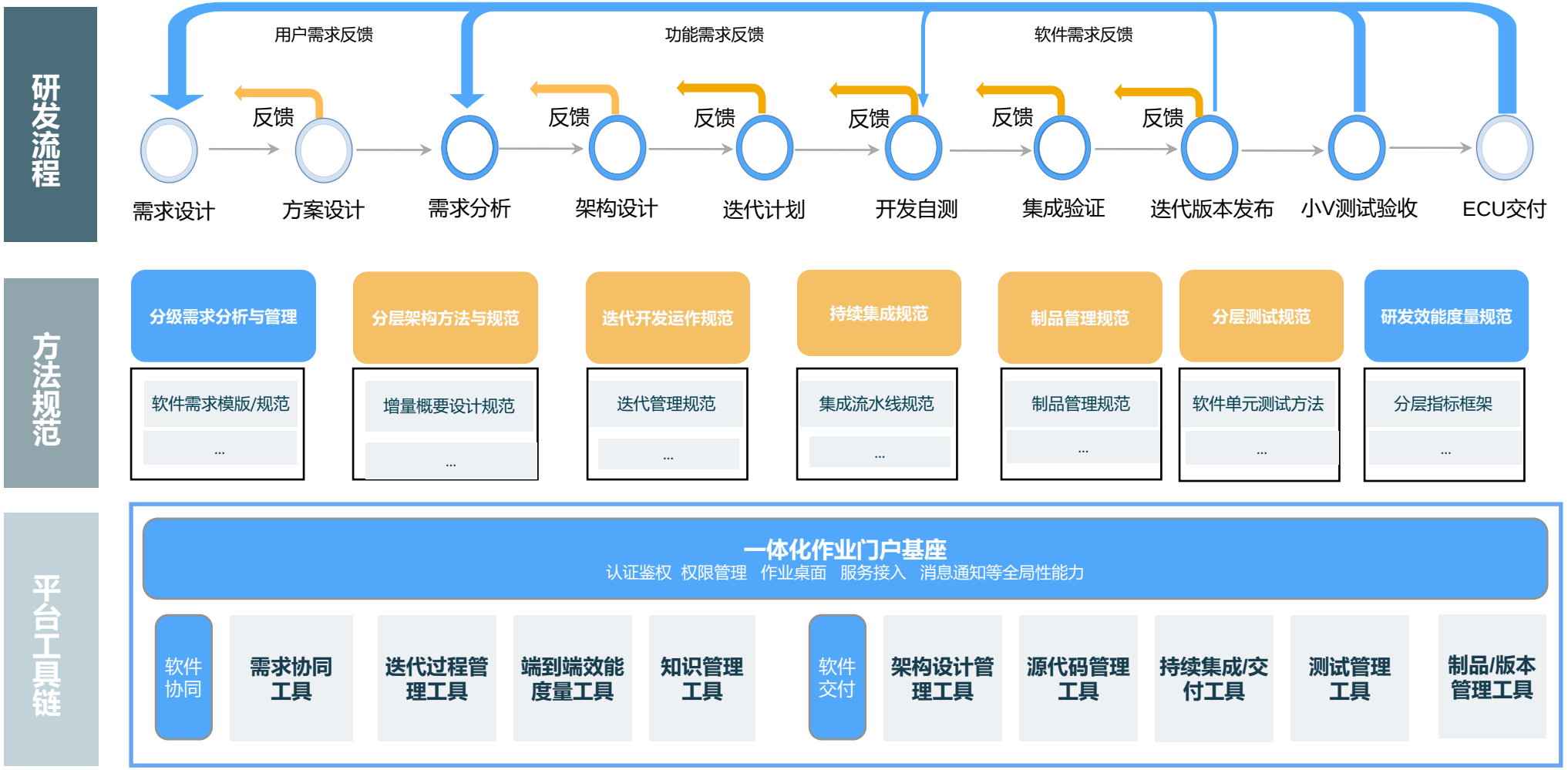


质量提升

降低错误率：
自动化流程减少了人为错误，提高了整体产品质量；

5、软件标准化研发能力体系建设

➢ DevOps是软件的标准化研发能力体系，通过流程、规范、工具进行系统性的改革、体系化的缩短反馈环，持续加速价值流动，以便用更低的成本保障软件持续高质量、快速、稳定、可控地交付。



5.1 一体化平台工具链

- 软件研发过程中的3大类工具：研发生产工具、研发支撑工具、研发协作工具在研发的不同阶段和环节中发挥着重要作用，合理的工具组合能够显著提升团队的工作效率和协作能力。
- 我们正在打造一个既能满足统一协作需求，又能适应各业务单元特性的研发协作平台。这一平台将显著提升整体研发效率，增强跨部门协作，并为管理决策提供强有力的数据支持。

研发支撑工具

支持研发交付各种支撑/辅助功能的工具，帮助团队更好地管理研发过程，
如缺陷管理工具、文档管理工具等

研发生产工具

研发交付各阶段不可或缺的关键性生产工具，直接用于软件开发和工程实现的工具，旨在提高代码质量、开发效率和产品交付速度。
如软件开发工具MATLAB/Simulink

研发协作工具

支持整体研发的关键业务作业的工具，促进团队内部及与其他团队之间的合作，
如沟通工具、知识共享平台等；

未来工作的思考与展望

DevOps转型带来的不仅是技术领域的变革，更是文化和管理理念的革命。通过持续的改进与学习，能够实现更高效的运作模式，推动产品与服务的创新。

我们将结合一体化工具平台和效能度量体系，以产品为导向，打造从需求到测试的端到端的软件研发链路，通过数据驱动决策，以**软件质量为第一目标**，确保产品的可靠性、可维护性和用户满意度，从而提升整体业务效益和市场竞争力。

安全是最大的吉利

2024 MathWorks
中国汽车年会

Thank you

MathWorks®