# 第一章

## 互联网时代的软件开发有哪些挑战？

1. **业务不断创新的挑战**，IT团队，应用和基础架构要能够灵活的支持业务的快速变革与尽快的切入市场。
2. **业务快速增长的挑战**，IT系统要能够支撑用户的快速增长。
3. **业务连续性的挑战**， 用户会随时随地地使用企业提供的服务，因此需要保证企业IT系统的24\*7可用

## 从五个层面来阐述什么是DevOps?

###### 价值观（DevOps宣言）

个体和互动 高于 流程和工具

工作的系统 高于 详尽的文档

客户及程序员合作 高于 合同谈判

响应变化 高于 遵循计划

###### 原则

1. 持续不断的及早交付功能（比“软件”更通用）
2. 软件功能只有在完整的系统交付给客户后才能实现。非功能性需求同样重要。（新增：为什么系统很重要）
3. 基础设施是代码，应该同样进行开发和管理。（新增。，）
4. 欣然面对需求变化，及时在项目后期。为了客户的竞争优势，敏捷过程掌控变化。（相同）
5. 较短周期交付可工作的功能（软件->功能）
6. 业务人员，开发人员和运维人员必须相互合作（添加运维人员）
7. 激发个体的斗志，以他们为核心搭建项目。提供所需的环境和支援，辅以信任，从而达成目标（相同）
8. 不论团队内外，传递信息效果最好效率最高的方式是面对面交谈。（相同）
9. 可工作的软件并进行完整交付是进度的首要度量标准
10. 敏捷过程倡导可持续开发。责任人，开发人员，运维人员和用户要能够共同维持其步调稳定延续。（添加运维人员）
11. 坚持不懈的追求技术卓越和良好设计，敏捷能力由此增强。（相同）
12. 以简洁为本，他是极力减少不必要工作的艺术。（相同）
13. 最好的架构、需求和设计出自组织团队。（相同）
14. 团队定期烦死如何能提高成效，并依此调整自身的举止表现。（相同）
15. 此外，精益的原则也得到广泛认可。
    1. 消除浪费
    2. 增强学习
    3. 尽量延迟决策
    4. 尽快交付
    5. 赋予团队权利
    6. 内建完整性
    7. 全局优化

###### 方法

常见的敏捷方法有 Scrum，XP，Kanban。

Scrum： 以经验过程控制为理论依据，采用迭代、增量的方法来提高产品开发的可预见性并控制风险。

极限编程： 把好的编程实践发挥到极致。

Kanban：主要规则有 可视化工作流； 限定 在制品； 衡量并管理周期时间。

###### 实践

DEVOPS中可以使用敏捷软件开发中常见的管理实现包括

迭代式计划、站立会议、回顾、评审、短周期迭代、团队估算等；

以及常见的技术实现包括

单元测试、持续交付、持续集成、编码标准、重构。

###### 工具

源代码库

构建服务器

配置管理

虚拟基础架构和容器

测试自动化

管道编排

# 第四章

## CMM/CMMI的五个成熟度级别的特征是什么？

1. CMM初始级（CMMI 初始级） 个人行为，软件开发过程混乱无序无准则指导。
2. CMM可重复级（CMMI 已管理级） 已建立基本的项目管理过程跟踪成本、进度和功能性；建立了必要的过程纪律，能重复利用以前的成功。
3. CMM已定义级（CMMI 已定义级） 软件过程文档化、标准化，并集成到组织的标准软件过程。项目活动是标准和一致的，不同项目采用相同的标准。
4. CMM已管理级（CMMI 定量管理级） 已经采用详细的有关软件过程和产品质量的度量，可有效预测。
5. CMM优化级（CMMI 优化级） 定量反馈，持续过程改进。

## 试比较CMMI和SPICE两个模型的异同。

|  |  |  |
| --- | --- | --- |
|  | CMMI | SPICE |
| 过程 | 过程管理；  项目管理；  工程；  支持； | 客户 – 供应商过程；  工程过程；  管理过程；  支持过程；  组织过程 |
| 级别 | 初始级；  已管理级；  已定义级；  定量管理级；  优化级 | 1. 不完整过程； 2. 已实施过程； 3. 已管理过程； 4. 已建立过程； 5. 可预测过程； 6. 优化过程； |

## DevOps模型对现有的软件开发过程和方法可能带来哪些影响？

目前并不存在专门为DevOps所定义的开发过程和方法，很多DevOps的实践者认为软件开发中的精益方法是DevOps的基础方法学。