

# 软件危机和软件过程

---

## 软件危机产生背景

Brooks的预测是千真万确的，软件危机的困境依然没有从根本上找到解决方法。

基于规则的编程模型，找不到解决软件危机的杀手锏（银弹）。

基于联结的编程模型，或许能在软件危机的困境中取得突破。

## 软件过程模型

描述性和说明性软件过程

### 瀑布模型

原型化瀑布模型 ~ 有利于确认需求是否被准确理解 和 验证技术方案的可行性

### V模型

将瀑布模型前后两端的过程活动结合起来，可以提高过程活动的内聚度。

生死相依原则

开始一项工作前，先思考验证该工作的方法。

### 分阶段开发

增量开发 ~ 从一个功能子系统开始，每次增加功能。

迭代开发 ~ 完成一个完整的系统，每次交付会升级其中的某个子系统

### 螺旋模型

引入了风险管理。在每个阶段构建原型是螺旋模型减小风险的基本策略。

每次迭代四个主要阶段

1. plan
2. determine goals, alternatives and constraints
3. evaluate alternatives and risks
4. develop and test

## 个人软件过程PSP

# 个体度量过程PSP0

- 一般来说，做任何事情最基本的思路就是计划、执行和总结。PSP0就是这个最基本的思路，但是在PSP0阶段必须理解和学会采集软件过程中的数据。
- 1、计划阶段
- 2、开发阶段
  - • 设计
  - • 编码
  - • 测试
- 3、开发完成后进行总结分析

# 个体度量过程PSP0.1

- PSP0.1 增加了编码标准规范、程序规模度量、以及过程改进计划。显然PSP0.1融入了标准规范，而且将最后的总结分析更加明确充实。过程改进计划用于随时记录过程中存在的问题、解决问题的措施以及改进过程的方法，以提高软件开发人员的质量意识和过程意识。
  - PSP0和PSP0.1重点是个体度量过程，也就是通过采集过程数据、度量程序的规模等，通过量化的度量数据为软件过程改进提供基准。
- 1、计划阶段
  - 2、开发阶段
    - • 编码标准规范
    - • 设计
    - • 编码
    - • 测试
  - 3、程序规模度量
  - 4、开发完成后进行总结分析
  - 5、过程改进计划

# 个体计划过程PSP1

- PSP1在计划阶段增加了项目评估，引入了基于估计的计划方法，用自己的历史数据来预测新程序的大小和需要的开发时间，开发阶段之后首先完成项目测试报告。
  - 在PSP1阶段应该学会编制项目开发计划，这不仅对承担大型软件的开发十分重要，即使是开发小型软件也必不可少。因为，只有对自己的能力有客观的评价，才能作出更加准确的计划，才能实事求是地接受和完成客户委托的任务。
- 1、计划阶段
    - • 项目评估
  - 2、开发阶段
    - • 编码标准规范
    - • 设计
    - • 编码
    - • 测试
  - 3、项目测试报告
  - 4、程序规模度量
  - 5、开发完成后进行总结分析
  - 6、过程改进计划

# 个体质量管理PSP2

- PSP2的重点是个体质量管理，引入了设计评审（Design Review）和代码评审（Code Review），以便及早发现缺陷，使修复缺陷的代价最小。
- 1、计划阶段
    - • 项目评估
  - 2、开发阶段
    - • 编码标准规范
    - • 设计
    - • 设计评审
    - • 编码
    - • 代码评审
    - • 测试
  - 3、统计记录各项工作用了多少时间
  - 4、项目测试报告
  - 5、程序规模度量
  - 6、开发完成后进行总结分析
  - 7、过程改进计划

# 个体质量管理PSP2.1

- PSP2.1引入了分析和设计规格，介绍了分析方法，并提供了设计规格。
- PSP2和PSP2.1的重点是个体质量管理，学会在开发软件的早期，发现由于疏忽所造成的程序缺陷问题。人们都期盼获得高质量的软件，但是只有高素质的软件开发人员并遵循合适的软件过程，才能开发出高质量的软件，PSP2引入并着重强调设计评审和代码评审技术，一个合格的软件开发人员必须掌握这两项基本技术。

- 1、计划阶段
  - • 项目评估
- 2、开发阶段
  - • 分析
  - • 设计规格
  - • 编码标准规范
  - • 设计
  - • 设计评审
  - • 编码
  - • 代码评审
  - • 测试
- 3、统计记录各项工作用了多少时间
- 4、项目测试报告
- 5、程序规模度量
- 6、开发完成后进行总结分析
- 7、过程改进计划

## 团队软件过程TSP

项目失败最根本的原因是团队问题

## 哪些常见的团队问题容易导致项目失败？

- • 缺乏有效的领导和管理；
- • 不能做出妥协、安排或不善于合作；
- • 缺少参与；
- • 拖拉与缺乏信心；
- • 质量低劣；
- • 功能多余；
- • 无效的组员互评；

# 团队的基本要素

- • 团队规模
- • 团队的凝聚力
- • 团队协作的基本条件

## 如何建设高效团队？

- • 建设具有凝聚力的团队
- • 设定有挑战性的目标
- • 反馈
- • 共同遵守的工作流程和框架

### **CMMI过程域**

- CMMI一级，初始级。没有包含任何过程域。
- CMMI二级，管理级。包含的过程域有需求管理 RM、项目计划PP、项目监督与控制PMC、供应协议管理SAM、过程与产品质量保证PPQA、配置管理CM和度量与分析 MA。

- **CMMI三级，已定义级。**包含的过程域有需求制定RD、技术方案TS、产品集成PI、验证VER、确认VAL、组织过程聚焦OPF、组织过程定义OPD、组织培训OT、集成项目管理IPM、风险管理RSKM、决策分析与决定DAR、集成供应商管理ISM、组织集成环境OEI和集成组队IT
- **CMMI四级，量化管理级。**包含的过程域有组织过程性能OPP和定量项目管理QPM。
- **CMMI五级，持续优化级。**包含的过程域有组织革新与部署OID和原因分析与决策CAR。

## 敏捷方法

个体和互动 高于 流程和工具

工作的软件 高于 详尽的文档

客户合作 高于 合同谈判

响应变化 高于 遵循计划

我们最重要的目标，是通过持续不断地及早交付有价值的软件使客户满意。

欣然面对需求变化，即使在开发后期也一样。为了客户的竞争优势，敏捷过程掌控变化。

经常地交付可工作的软件，相隔几星期或一两个月，倾向于采取较短的周期

业务人员和开发人员必须相互合作，项目中的每一天都不例外

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。

不论团队内外，传递信息效果最好效率也最高的方式是面对面的交谈。

可工作的软件是进度的首要度量标准。

敏捷过程倡导可持续开发。责任人、开发人员和用户要能够共同维持其步调稳定延续

坚持不懈地追求技术卓越和良好设计，敏捷能力由此增强

以简洁为本，它是极力减少不必要工作量的艺术

最好的架构、需求和设计出自自组织团队

团队定期地反思如何能提高成效，并依此调整自身的举止表现

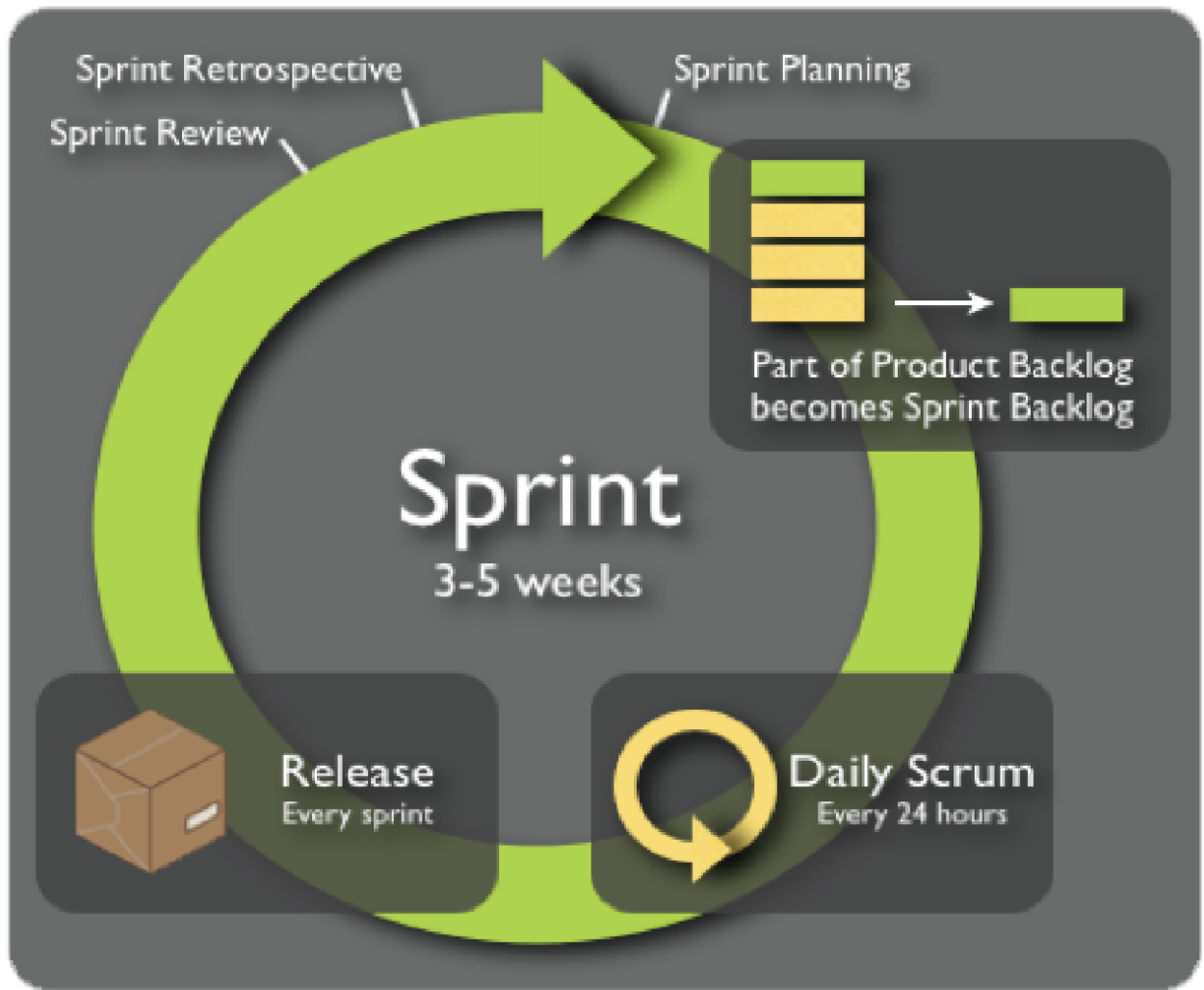
## Scrum敏捷开发方法

1. 项目经理
2. 产品经理
3. 团队

每一轮迭代是一个冲刺，包含有

1. 冲刺规划会议
2. 每日站立会议
3. 冲刺评审会议

## 4. 冲刺回顾会议



## 工作和跟踪进展

1. 产品积压订单
2. 冲刺积压订单
3. 燃尽图
4. 障碍积压订单

## 冲刺计划会议要点

会议要有足够时间，最好至少8小时。

取出部分产品积压订单做成Sprint积压订单，写成索引卡。

确定并细分每一个索引卡的故事。

确定每日站会的时间和地点。

确定好演示会议和回顾会议的日期。

## 每日站立会议要点

10-15min

昨天做了什么 今天要做什么 遇到了什么问题

使用好任务看板 更新燃尽图

## 冲刺评审会议要点

让老板和客户看到演示效果

不要关注太多细节，以主要的功能为主

### 冲刺回顾会议讨论的问题举例

1. 应该花更多的事件，把故事拆分成更小的条目和任务
2. 办公室环境太吵了
3. 做出过了过度的承诺，最后只完成了一半工作

### Scrum基本流程

step1: 找出完成产品需要做的事情。每一项时间估计单位为“天”

step2: 决定当前的冲刺。每一项单位为“小时”

step3: 冲刺。

step4: 得到一个软件的增量版本，冲刺评审会议，发布给用户。

### Scrum主要缺陷

1. 压力大
  2. 程序维护成本高
  3. 无法被中断
- 改善：
- 结合XP
- 测试驱动开发TDD
- 40小时工作制
- 使用编码规范

## DevOps

DevOps是一组过程、方法与系统的统称。可以看作开发、技术运营和质量保障三者的交集。

“一套旨在缩短从提交变更 到 变更后的系统投入正常生产之间时间，同时确保产品高质量的实践方法”

可以看作是敏捷方法从技术开发领域扩展到业务运维领域。

**精益生产**

**精益创业**

**最小可行产品MVP**