

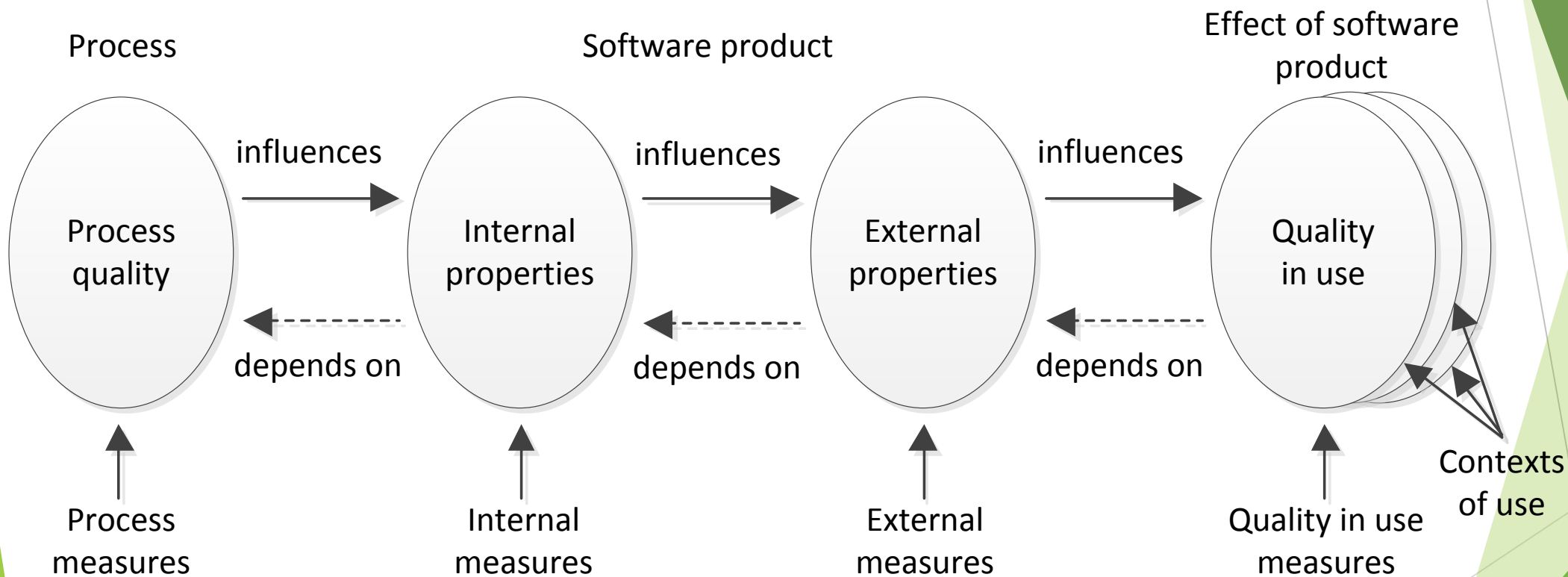
# 软件质量度量—— 过程中质量



# 概述

- ▶ 产品质量度量
- ▶ 过程中质量度量
- ▶ 软件维护中质量度量

# 软件生命周期中的质量



# 常见的过程中度量方法

- ▶ 机器测试阶段的缺陷密度
- ▶ 机器测试阶段的缺陷到达模式
- ▶ 基于阶段的缺陷移除模式
- ▶ 缺陷移除效率

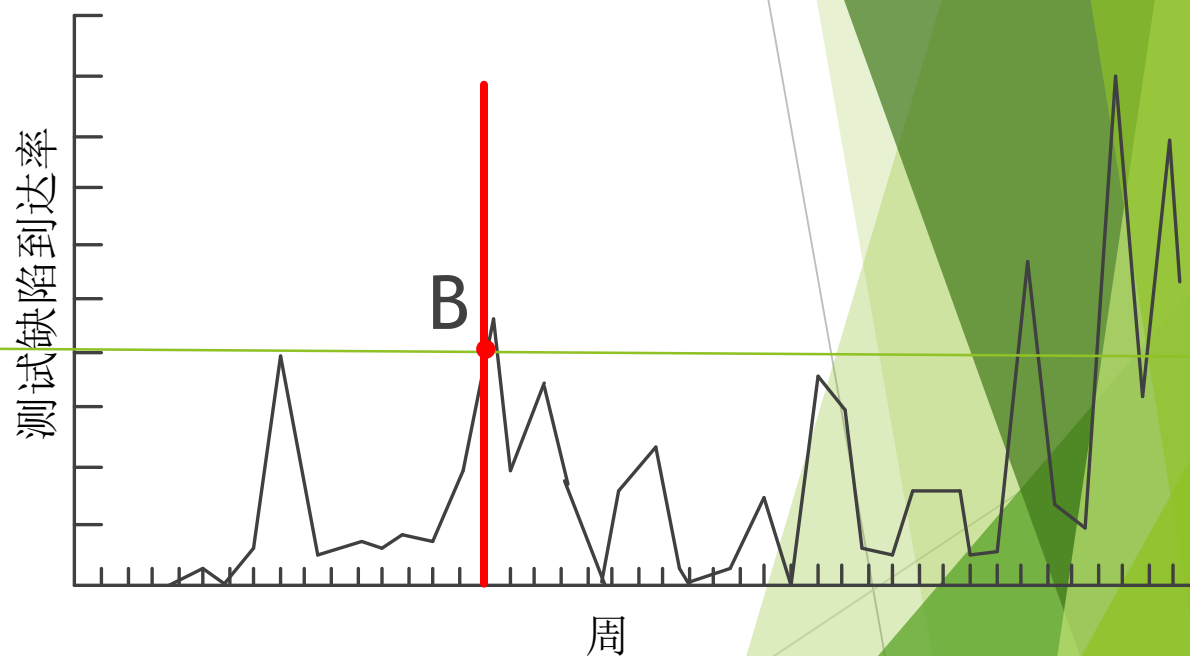
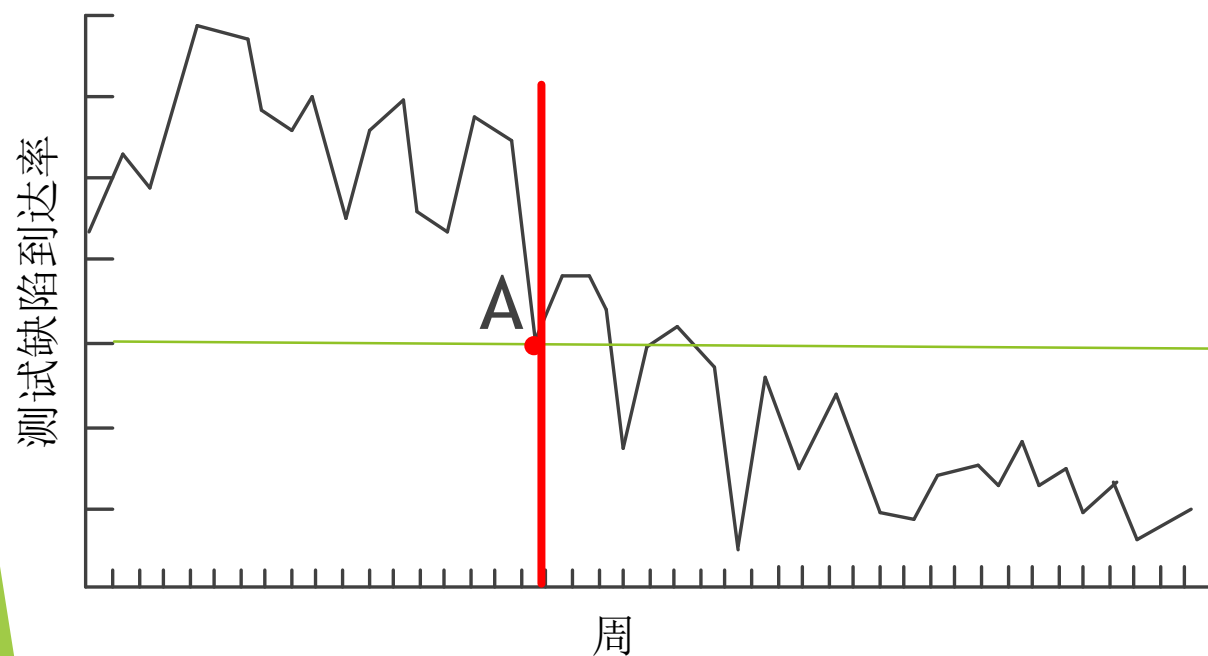
# 1. 机器测试阶段的缺陷密度 ( Defect density )

- ▶ 机器测试阶段的缺陷率通常与软件产品在实际应用中的缺陷率成正比。
- ▶ 缺陷率越**低**，过程质量越**高**
- ▶ 若测试阶段缺陷率**低于**前一个版本的缺陷率
  - ▶ 若并未提高测试效率，则显然过程**质量提高了**
  - ▶ 若已提高了测试效率，则需进一步测试才能判断
- ▶ 若测试阶段缺陷率**远远高于**前一个版本的缺陷率
  - ▶ 若并未提高测试效率，则显然过程**质量下降了**
  - ▶ 若已提高了测试效率，则可能质量并无变化

## 2. 机器测试阶段的缺陷到达模式 ( Defect arrival mode )

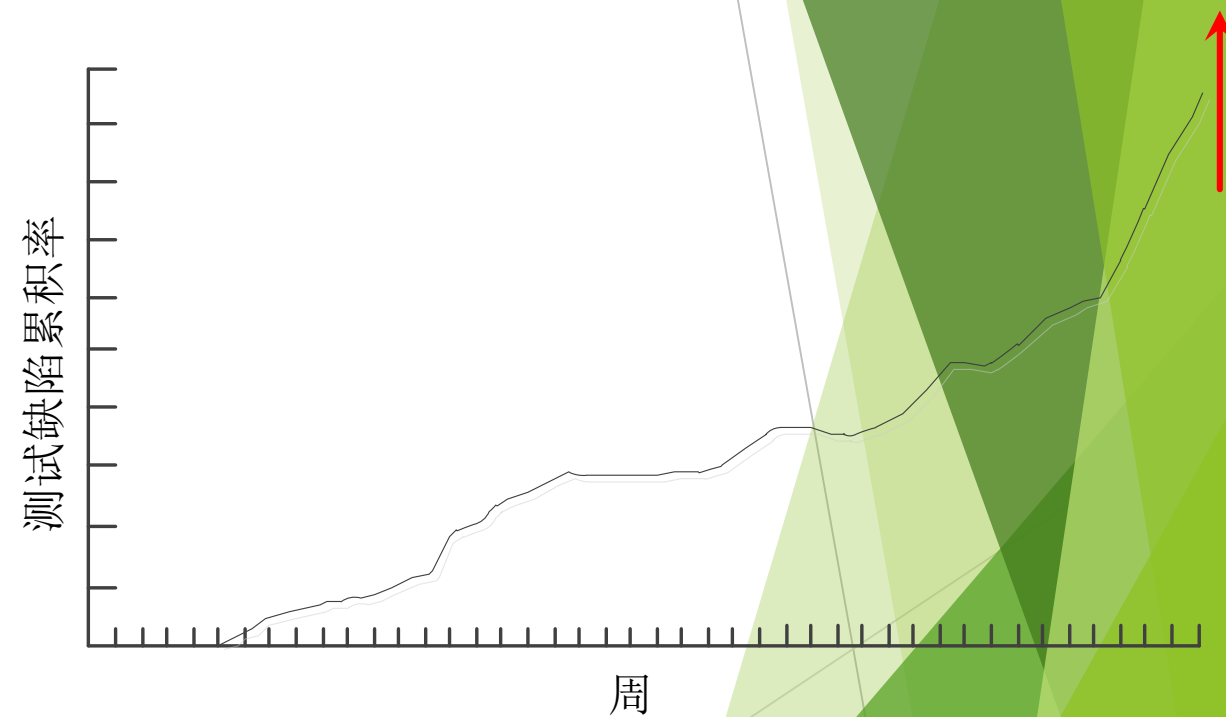
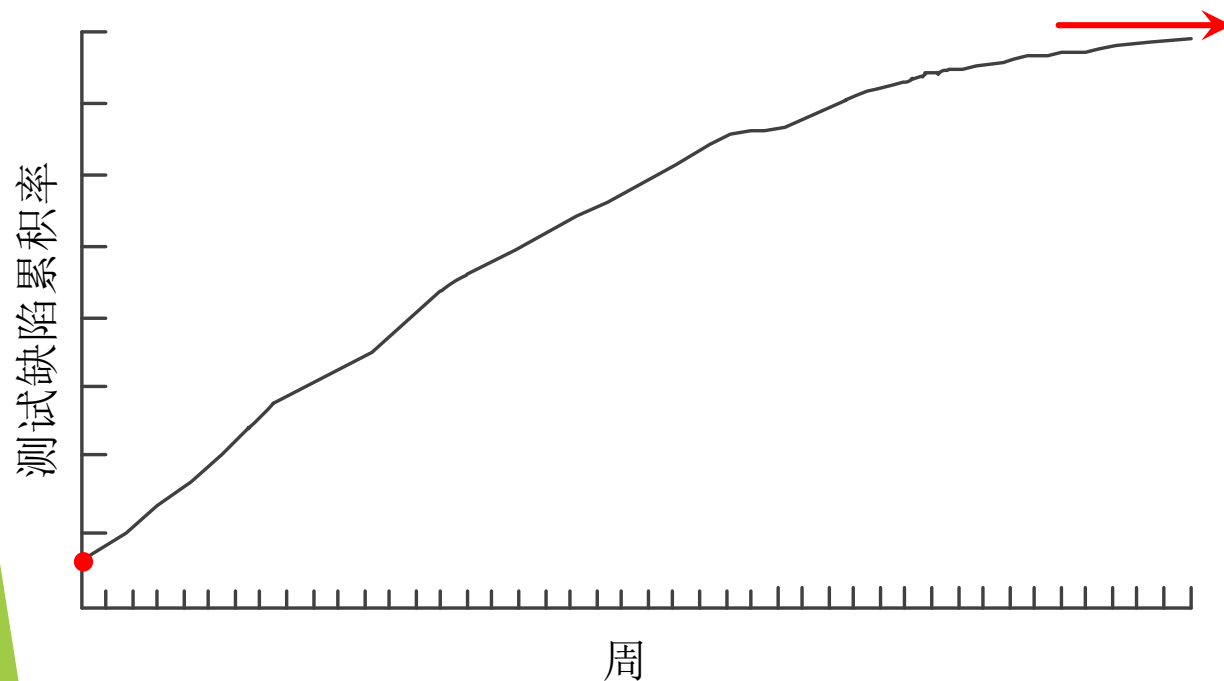
- ▶ 一段时间内缺陷率随时间变化的趋势
- ▶ 目标：在发布之前的测试阶段试图找到一种缺陷到达模式，使之稳定在一个较低的程度，或使缺陷率随时间推移而呈下降趋势。
- ▶ 常见的度量
  - ▶ 用时间间隔测量的缺陷到达模式
  - ▶ 有效缺陷的到达模式
  - ▶ 缺陷随时间积累的模式

# 缺陷到达率模式



测试阶段两种缺陷到达模式的对比

# 缺陷累积模式



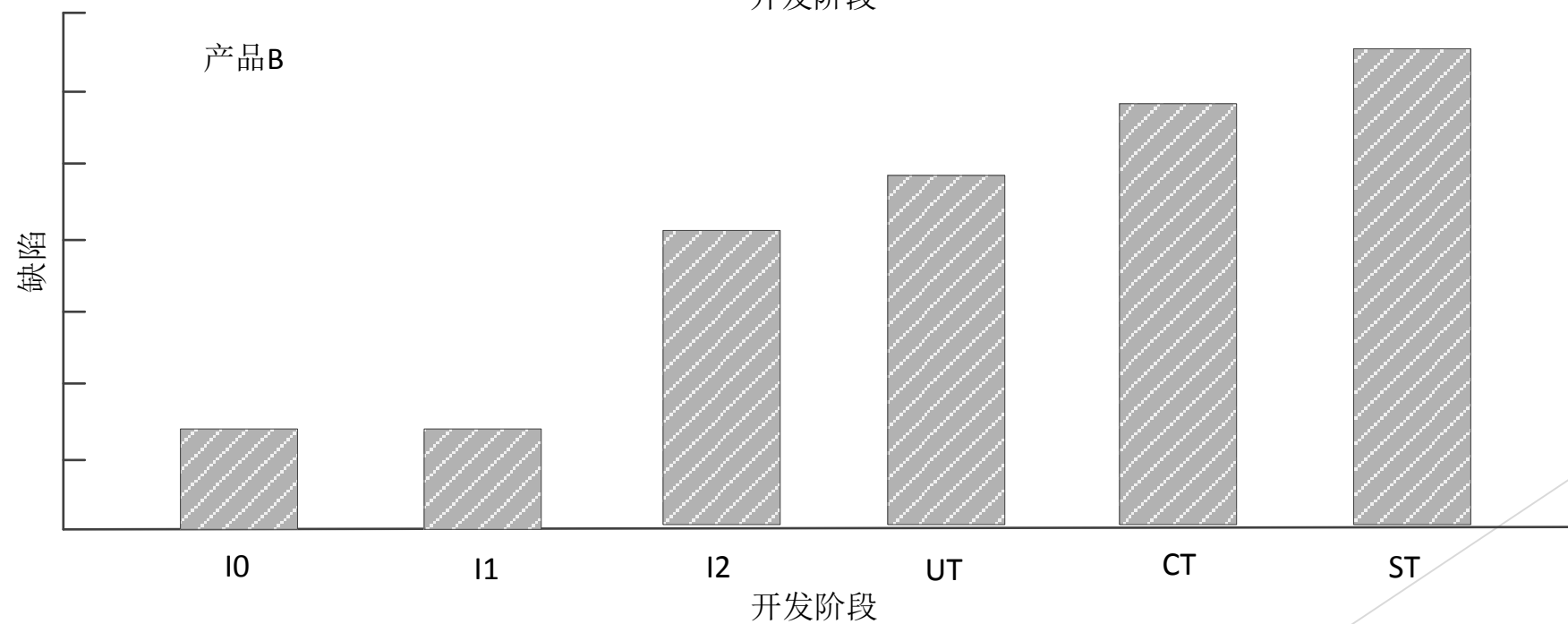
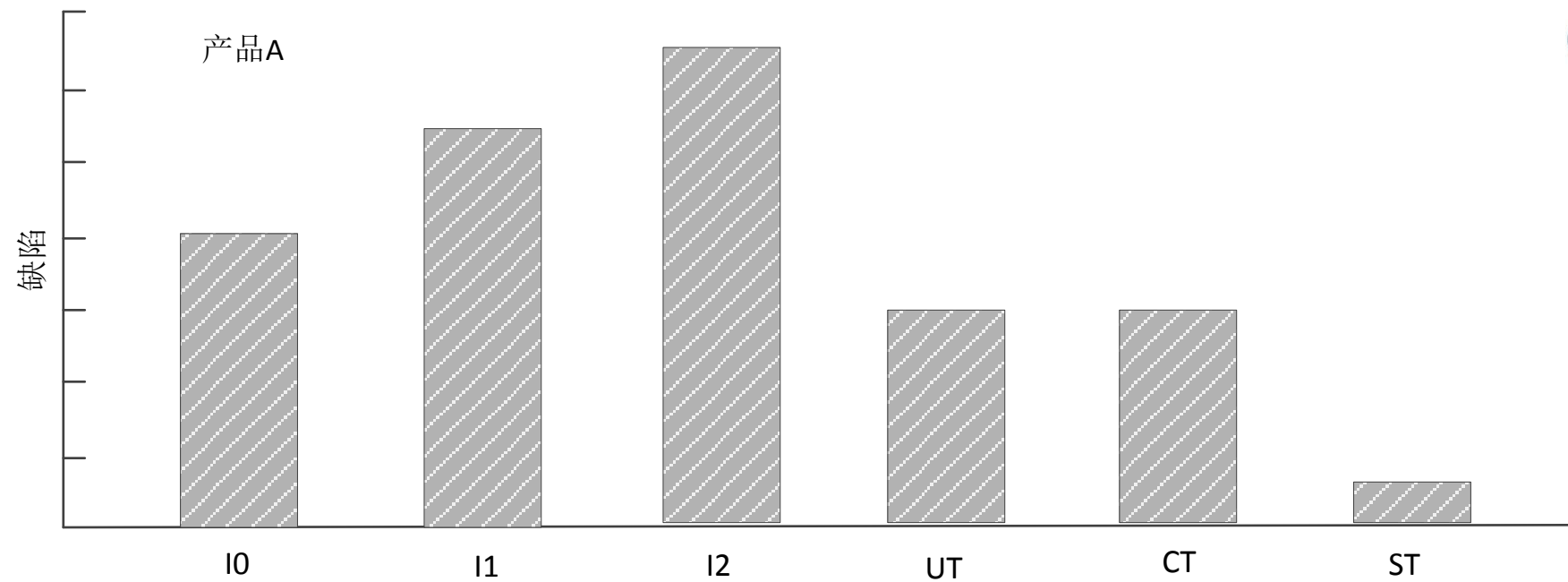
测试阶段两种缺陷累积模式的对比





### 3. 基于阶段的缺陷移除模式 ( Defect removal mode )

- 反映整个开发过程中的全面缺陷移除能力



## 4. 缺陷移除效率 ( Defect removal effectiveness, DRE )

- ▶ 阶段缺陷移除效率 = 本阶段发现的属于本阶段功能的缺陷总数 / 产品中潜在的本阶段缺陷总数
- ▶ 产品中潜在的本阶段缺陷总数  $\approx$  本阶段已发现的缺陷数 + 以后发现的缺陷数
- ▶ 整体缺陷移除效率  $\approx$  发布前发现的缺陷数 / (发布前发现的缺陷数 + 发布后发现的缺陷数)

## 4. 缺陷移除效率 ( Defect removal effectiveness, DRE )

- ▶ 设某软件系统发布前发现缺陷数:  $D1$ , 发布后发现缺陷数:  $D2$ ,
- ▶ 则发现的缺陷总数:  $D = D1 + D2$
- ▶ 整体缺陷移除效率  $\approx D1 / (D1 + D2)$



	需求分析	设计	编码	测试
需求评审	R1			
设计评审	R2	D1		
编码	R3	D2	C1	
测试	R4	D3	C2	T
发布	R5	D4	C3	N/A
缺陷总计	$R=R1+R2+R3+R4+R5$	$D=D1+D2+D3+D4$	$C=C1+C2+C3$	T
缺陷移除效率	R1/R	D1/D	C1/C	100%