

《度量信息系统交付质量》

图书基本信息

书名：《度量信息系统交付质量》

13位ISBN编号：9787121160882

10位ISBN编号：7121160889

出版时间：2012-3

出版社：电子工业出版社

作者：亚文.W.德根

页数：331

译者：赵皓

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《度量信息系统交付质量》

内容概要

《度量信息系统交付质量》针对软件分析、设计、测试、维护等各个阶段，研究提高软件可信性的理论和方法，研究如何对软件可信性进行评估和度量。全书共分为十五章，具体包括：信息系统质量概述、软件质量概念和管理问题概述等。

《度量信息系统交付质量》

作者简介

作者:(美)Duggan

书籍目录

第一部分 质量概念引论和概述

第1章 信息系统质量概述

- 1.1 绪论
 - 1.2 信息系统交付困难的本质
 - 1.2.1 技术上面临的挑战
 - 1.2.2 信息系统交付生命周期中的挑战
 - 1.3 信息系统质量模型
 - 1.4 信息系统质量的人员维度
 - 1.5 信息系统质量的过程管理维度
 - 1.5.1 cmm和cmmi
 - 1.5.2 建立信息系统开发过程
 - 1.5.3 软件生产方法
 - 1.6 信息系统产品质量和成功的系统
 - 1.6.1 产品质量属性
 - 1.6.2 信息系统质量与信息系统成功
 - 1.7 总结与结论
- 参考文献

第2章 软件质量概念和管理问题概述

- 2.1 引言
 - 2.2 软件质量基础
 - 2.2.1 软件工程文化与伦理
 - 2.2.2 质量的价值和成本
 - 2.2.3 模型和质量特征
 - 2.3 软件质量管理过程
 - 2.3.1 软件质量保证过程
 - 2.3.2 软件产品质量
 - 2.4 实践考虑
 - 2.4.1 软件质量需求
 - 2.4.2 软件质量测量
 - 2.4.3 软件质量的改进
 - 2.5 未来趋势和小结
- 参考文献

第3章 软件质量观

- 3.1 引言
 - 3.2 背景
 - 3.3 质量的含义
 - 3.3.1 各种质量观
 - 3.3.2 软件质量观
 - 3.3.3 模型评论
 - 3.3.4 更多关于产品模型的研究
 - 3.4 用于测度软件质量的软件评价框架
 - 3.5 总结
- 参考文献

第二部分 信息系统交付早期的质量

第4章 一种需求缺陷问题的解决方法

- 4.1 简介
- 4.2 行为树

4.3 演化设计

4.3.1 前置条件公理

4.3.2 交互公理

4.4 需求翻译

4.4.1 实例翻译

4.5 需求集成

4.5.1 集成实例

4.5.2 集成缺陷检测

4.5.3 缺失条件和事件

4.5.4 逆转缺陷的缺失

4.5.5 死锁、活锁及安全性检查

4.5.6 逆转矛盾缺陷的发现

4.6 与其他方法的比较

4.7 结论

参考文献

第5章 用户参与需求规约质量保证

5.1 简介

5.2 用户参与改进需求规约质量

5.3 用于规约的模型概述

5.3.1 流程图

5.3.2 实体关系图 (erd)

5.3.3 决策表

5.4 统一建模语言(uml)用例

5.4.1 petri网

5.5 动画系统工程 (ase)

5.5.1 ase有效性指标

5.6 总结和结论

参考文献

第6章 学术故事

6.1 引言

6.1.1 背景

6.2 论点和问题

6.2.1 确认

6.2.2 场景关注点

6.3 解决方法和建议

6.3.1 确认

6.3.2 场景方法学

6.3.3 场景范例

6.3.4 设备：农夫助手

6.3.5 性能

6.4 情景

6.4.1 加油：一次昂贵的测试

6.4.2 收割作物

6.5 结论

参考文献

致谢

第三部分 过程对信息系统质量的贡献

第7章 过程中心论对信息系统质量的贡献

7.1 引言

7.2 背景：过程管理的有效性

7.2.1 能力成熟度模型

7.2.2 成熟度等级

7.3 信息系统过程管理：原则和实践

7.3.1 开发人员

7.3.2 技术

7.3.3 方法

7.4 对过程中心论的展望

7.4.1 公认的优势

7.4.2 报告的问题

7.4.3 折中观点

7.5 启示与结论

参考文献

尾注

第8章 开发者驱动的信息系统质量——软件过程改进的指导方针

8.1 引言

8.1.1 研究目的

8.1.2 研究问题

8.2 背景和研究基础

8.2.1 在组织中定义成功的spi推广

8.2.2 影响spi满意度的因素：个人控制

8.2.3 影响spi使用的因素：质量、生产力、对有效性的认知

8.3 研究方法

8.3.1 将个人软件过程（pspsm 1）作为软件过程改进实例

8.3.2 数据收集

8.4 讨论

8.4.1 开发者个人控制

8.4.2 对质量和生产力的体会

8.4.3 有效性认知的其他因素

8.5 选择和执行spi的管理指导方针

8.5.1 选择spi的指导方针

8.5.2 spi培训的指导方针

8.5.3 选择执行环境的指导方针

8.6 未来的研究方向

参考文献

尾注

第9章 系统开发中使用敏捷方法提高质量

9.1 引言

9.2 质量定义

9.3 系统开发的其他方法

9.3.1 原型和快速应用开发

9.3.2 其他方法与sdlc的不同之处

9.4 敏捷方法

9.4.1 敏捷方法的原则和价值

9.4.2 敏捷方法可视为一种哲学思想

9.4.3 敏捷核心实践

9.5 结对编程：敏捷方法实践的实例

9.5.1 程序设计的团队方法

9.5.2 结对编程的优点

- 9.5.3 结对编程的缺点
- 9.5.4 敏捷方法的价值观
- 9.5.5 价值观如何塑造项目开发
- 9.5.6 接受改变很重要
- 9.5.7 资源权衡是可能的
- 9.5.8 保证质量是关键
- 9.5.9 使用敏捷方法需要注意的问题
- 9.6 未来的研究问题
- 9.7 结论和建议

参考文献

第10章 质量度量和贝叶斯分析——极限编程案例

- 10.1 引言
- 10.2 背景
 - 10.2.1 xp相关研究文献
 - 10.2.2 研究假设
- 10.3 实验内容和设计
 - 10.3.1 项目参与者和过程
 - 10.3.2 实验设计
 - 10.3.3 变量和测度
 - 10.3.4 分析过程
- 10.4 分析结果
- 10.5 讨论
- 10.6 总结

参考文献

第四部分 软件过程改进计划和方法学的风险管理

第11章 信息系统风险管理方法

- 11.1 引言
- 11.2 风险管理文献分析
 - 11.2.1 风险列表
 - 11.2.2 风险-行动列表
 - 11.2.3 风险-策略模型
 - 11.2.4 风险-策略分析
- 11.3 行动研究
- 11.4 案例：spi中的风险管理方法
 - 11.4.1 企业案例
 - 11.4.2 行动研究计划
 - 11.4.3 管理软件过程改进风险
- 11.5 结论

参考文献

第12章 检查信息系统开发方法学中评价框架和元建模范型的质量

- 12.1 介绍
- 12.2 元模型、评价框架和元信息管理
 - 12.2.1 基本概念
 - 12.2.2 元模型和评价框架使用的基本原理
- 12.3 方法评价框架和方法工程元模型的检验和粗略分类
 - 12.3.1 欧洲方法
 - 12.3.2 一种通用评估框架
 - 12.3.3 desmet
 - 12.3.4 tudor及其框架

- 12.3.5 社会控制论框架
- 12.3.6 nimsad
- 12.3.7 metaphor
- 12.3.8 case?shells
- 12.3.9 metaview系统
- 12.3.10 case数据交换格式 (case data inferchange format,cdif)
- 12.3.11 cdm?filters
- 12.4 其他方法元建模和方法集成项目
- 12.5 评价框架和元模型总结
- 12.6 已知框架和元模型剖析
- 12.7 结论
- 参考文献

第五部分 新兴领域中的信息系统质量问题

第13章 软件质量和开源过程

- 13.1 介绍
- 13.2 背景
 - 13.2.1 软件质量
 - 13.2.2 开源过程
 - 13.2.3 专利软件与开源软件
- 13.3 开源机制
 - 13.3.1 软件的生产者
 - 13.3.2 软件使用者
 - 13.3.3 同行评审过程
 - 13.3.4 调试和反馈
 - 13.3.5 演化式生存
- 13.4 质量和开源过程
 - 13.4.1 用户是共同开发者
 - 13.4.2 早发布、常发布
 - 13.4.3 给予足够关注
- 13.5 软件方法学比较
- 13.6 未来趋势
- 13.7 结论
- 参考文献
- 尾注

第14章 创建政府部门信息系统质量

- 14.1 引言
- 14.2 背景：公共信息系统的不同之处
- 14.3 多变的政府质量需求
 - 14.3.1 政府系统质量需求因设计而不同
 - 14.3.2 高质量系统经久耐用；合格的质量系统是埋头苦干的老黄牛
 - 14.3.3 政府信息系统不同质量的案例
- 14.4 未来趋势：没有银弹
- 14.5 结论
- 参考文献

第15章 企业资源规划 (erp) 质量

- 15.1 介绍
- 15.2 erp进入高等教育
- 15.3 erp文献
 - 15.3.1 大学部门

《度量信息系统交付质量》

- 15.4 质量相关的文献
 - 15.4.1 质量的用户视角
- 15.5 研究方法
 - 15.5.1 框架
 - 15.5.2 方法
- 15.6 案例研究
 - 15.6.1 需求质量
 - 15.6.2 接口质量
- 15.7 讨论
- 15.8 结论

序曲

参考文献

关于作者

术语表

章节摘录

版权页：插图：前面的两种缺陷同时发生。解决这类型的问题有时需要领域知识。在某些情况下，当我们试图集成一个RBT时，我们发现并不是只有叶子节点会与其他RBT或局部DBT产生交迭。在这些情况下，这种冗余可以在集成时被排除。虽然从原理上讲，构造一个算法使得集成步骤“自动化”是可能的，虽然，现在已有支撑工具可以用来识别满足集成匹配准则的节点。但是集成问题是我们实际系统中经常碰到的问题，因此如果能有人参与控制整个集成过程效果会更好。我们在大型工业系统中进行集成的经验是，这种方法能够及早发现用传统形式化审查方法发现不了的一些问题。我们获得经验是，需求集成是一种关键的完整性检查，所以对用于构建设计基础的需求集进行完整性检查是明智的。审查及自动化检测缺陷一旦我们拥有一组被表示为集成设计行为树的功能需求，我们就有了实施缺陷检测步骤的有利条件。设计行为树证明是一种十分有效的表示法，它能反映出一系列在初始需求描述中常见的、不完全的及不一致的缺陷。微波炉系统案例研究中提到了不完整性及其他缺陷。DBT可以进行人工书面形式化审查，但由于行为树拥有形式化语义（Winter，2004），因此我们也可以使用工具（Smith等，2004）进行自动化的形式分析。这些工具的组合为发现缺陷提供了一个强有力的武器。对简单的实例来说，如微波炉系统，我们很容易进行书面审查并识别大量的缺陷。对较大的系统，即拥有大量状态及复杂的控制结构的系统，自动化工具对系统的、基于逻辑、可重复的缺陷发现十分重要。我们现在认识到很多有条不紊的人工和自动缺陷检查能够基于DBT开展。

4.5.3 缺失条件和事件 一个常见问题是关于初始需求描述方面的。初始需求描述了可能应用在系统行为中的某一点的一组条件，其中通常会遗漏促使行为完整的情况。这种最简单的情况是：在需求中的某个位置说明了如果运用某一条件应该会发生什么，但其中并没有记载如果没有运用这个条件会发生什么。在系统行为的某一点中也可能会存在缺失的事件。

《度量信息系统交付质量》

编辑推荐

《度量信息系统交付质量》可作为从事军事信息系统建设的技术人员及管理人员学习的参考书。

《度量信息系统交付质量》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com