Nickname: mike6606
Age: 8 years 10 months
Fans: 0
Followers: 5
+ Plus follow

< March 2023 >

| day | One | Two | Three | Four | Five | Six |
|---|---|---|---|---|---|---|
| 26 | 27 | 28 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 31 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Search

Look for it

Google search

Frequently used links

My essays
My comment
My involvement
Latest comments
My labels

Essay archives

January 2021 (1)
January 2018 (3)
January 2017 (8)

Read the leaderboard

1. Software Engineering Practitioners' Research Methods Chapter 13 Answers (1578)
2. Software Engineering Practitioners' Research Methods Chapter 1471 Answers (<>)
3. Software Engineering Research Methods for Practitioners Chapter 1290 Answers (<>)
4. Software Engineering Practitioners' Research Methods Chapter 34 Answers (1282)
5. Software Engineering Practitioners' Research Methods Chapter 1264 Answers (<>)

Recommended leaderboards

1. Software Engineering Practitioners' Research Methods Chapter 1 Answers (<>)

## Software Engineering Practitioners' Research Methods Chapter 19 Answers

Problem:

Describe how you would assess the quality of a university before applying to it. What factors would be important? Which would be critical?

Answer:

4633-14-1P SA: 9420

SR: 6376

The quality can be assessed by observing various aspects. The important factors to be observed are:

1. **Standards:** The standards that the university is maintaining and whether it is following all the applicable standards or not.

2. **Recognition:** How good the university is certified by the government and recognition it has.

3. **Quality of teaching:** How good the teaching is? And how the results are obtained?

4. **Infrastructure:** Check whether a good and supportive environment is present for students to concentrate well on studies or not.

5. **Behavior and Discipline:** Is the management well concerned about teaching well disciplined activities and behavio and also making the students to implement them.

Of all the factors the most critical are the standards and the quality of teaching and knowledge being provided.

Problem:

Garvin [Gar84] describes five different views of quality. Provide an example of each using one or more well-known electronic products with which you are familiar.

Answer:

According to David Garvin[Gar84] definition of Quality is a complex and multifaceted concept. Garvin defined the quality in five different views.

**They are as follows:**

- Transcendental view
- Users view
- Manufacture’ s view
- Product view
- Value-based view

**Transcendental view:**

It is defined as the metaphysical aspect of quality, which is nothing but, something that immediately recognizes, that cannot explicitly define. Transcendental view also defined as “I can’ t’ define, but I know when I see it”

**Example:**

When the usage of cosmetics or daily products, the statement is like “I love this product” , I felt very beautiful. So, it is the external feature that defines the product for quality.

**User view:**

User view quality is shortly defined as “fitness for intended use” . It is defined as if the product can meet the expectations of the end-user goals. Then it exhibits the quality. For example, customer-based expectations are a wide variety of interests, needs, specific quality standards, and other external features. Hence, user view quality mainly depends on how to fulfill customer needs.

**Example:**

Product doesn’ t deal only the delivery of the product. It depends on the service of the product. Nissan offering ‘dud’ models in US markets under the brand name Datson which the US customer didn’ t prefer.

manufacturing practice, or conformance to specification.” . Manufacture based view defined as the original specification of the product. The manufacture-based view, the ultimate outcome is the total customer satisfaction, that not only involves external customers but internal ones as well.

**Example:**

Cheese – “quality is about manufacturing a product that people can depend on every time they reach for it.” Any drinking products are eating products are examples of manufacture-based quality.

**Product view:**

It is defined as the product quality based on inherent functions and features. This product doesn’ t have an impact on customer’ s individual views or opinions. The product-based scenario based on some hard data of the product.

**Example:**

Any electrical products, e.g. how fast it restarts, or how much time left between failures, etc.

**Value-Based View:**

The Value-Based Perspective defines quality in a product as one that is as useful as competing products and is sold a a lesser price. Value-Based quality mainly deals with the cost and price of the product. Quality is all the features that allow a product to satisfy stated or implied needs at an affordable cost.

**For example:**

US auto market Incentives offered by the Big Three are perceived to be compensation for lower quality.

Problem:

Using the definition of software quality proposed in Section 19.2, do you think it’ s possible to create a useful product that provides measurable value without using an effective process? Explain your answer.

Answer:

4633-14-3P SA: 9420

SR: 6376

**No,** it is not possible to create a useful product that provides measurable value without using an effective process. Because,

• Software engineering practices allow the developer to analyze the problem and design a solid solution to build high-quality software.

• Umbrella activities such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.

Problem:

Add two additional questions to each of Garvin’ s quality dimensions presented in Section 19.2.1.

Answer:

4633-14-4P SA: 9420

SR: 6376

David Garvin describes eight dimensions of quality.

**Performance quality:**

1. Does the software contain main operating features such as power, clarity, speed etc.?

2. Does the software requirements are easy to assess by the user?

**Feature quality:**

1. Does the software contain the extras that supplement the main characteristics such as stylish, sunroof etc.?

2. Does the software provide new special features

**Reliability:**

1. How often it breaks down?

2. Does the software continuously run well?

**Conformance:**

1. How close it is to the design specification?

2. Does it provide a minimum guaranteed quality?

2. Can it accommodate the changes?

**Serviceability:**

1. Does the software have the ease to be repaired?

2. Does the quality software be available at low-cost?

**Aesthetics:**

1. Is the software understandable?

2. Is it user-friendly software?

**Perception:**

1. Is it acceptable and likely?

2. How are the feel, finish and manner?

Problem:

McCall’s quality factors were developed during the 1970s. Almost every aspect of computing has changed dramatically since the time that they were developed, and yet, McCall’s factors continue to apply to modern software. Can you draw any conclusions based on this fact?

Answer:

McCall’s quality factors are

- Correctness
- Reliability
- Efficiency
- Integrity
- Usability
- Maintainability
- Flexibility
- Portability
- Reusability
- Interoperability

The conclusions that can be drawn are:

- These factors are like a basement. They provide the basic rules for determining the quality of software.
- Though the quality factors were developed during the 1970’s they are not designed for any particular software or product.
- For any software they are the common and general factors to be noted, to determine the quality. So they remain unchanged.
- They are some generalized factors and can be applied to any product of any generation. Hence they continue to apply even to the modern software.

Problem:

Using the subattributes noted for the ISO 9126 quality factor “maintainability” in Section 19.2.3, develop a set of questions that explore whether or not these attributes are present. Follow the example shown in Section 19.2.4.

Answer:

The sub-attributes of the quality factor maintainability are:

- Analyzability
- Changeability
- Stability
- Testability

The questions that explore whether or not these attributes are present are as follows:

**Analyzability –** The ease with which the cause of a failure can be determined.

- Are the modules open for extension?

**Changeability –** The ability to accommodate changes.

- Is the modification of the design more difficult?

- Is the design less understandable?

**Stability –** The risk of a modification.

- Does the software functions continuously well?

- Is the risk of a modification to the software low?

**Testability –** Testing the quality and functionality of the product

- Does the product meet the requirements?

- Does it match the specification?

- Is the design complex?

- Does it work as expected?

Problem:

Describe the software quality dilemma in your own words.

Answer:

4633-14-7P SA: 9420

SR: 6376

**Software quality dilemma** is the dilemma of deciding the quality that a software product must contain. Every software organization is faced with the software quality dilemma.

If software provided low quality then no one will want to buy. So basically, everyone wants to build high-quality systems. But it needs more time and more effort to produce “perfect” software, which is simply unavailable in a market-driven world. It may happen that the product may not have launched in time and it may not acquire recognition. Also the resources and the effort applied will be wasted. This is the dilemma of deciding the quality that is needed while developing a software.

Problem:

What is “good enough” software? Name a specific company and specific products that you believe were developec using the good enough philosophy.

Answer:

4633-14-8P SA: 9420

SR: 6376

**Good enough software** is the software that delivers high-quality functions and features for user needs that has sufficient quality to satisfy the customer’ s requirements. At the same time it delivers other more obscure or specialized functions or features that contain known bugs.

Microsoft is the company that develops its products using good enough software. It develops a basic version initially and releases it, satisfying the needs and the quality required by customers. Later it develops the enhanced versions and releases.

The products it developed using good enough software include Microsoft Office PowerPoint, Microsoft Office Word, and Microsoft Windows etc.

- Microsoft office 2010 is the recently released version and its previous versions are Microsoft office 2003, Microsoft office 2007.

- Microsoft Office word was released in different versions as, Word 2003, 2007.

- Similarly Microsoft Windows was released in different versions. They include Windows 95, Windows NT, Windows XI Windows Vista, and Windows 7 etc. These all are produces using the concept of good enough software.

Problem:

Considering each of the four aspects of the cost of quality, which do you think is the most expensive and why?

Answer:

The four aspects involved in the cost of quality are:

- Internal failure costs
- External failure costs

**Prevention costs:** The costs of all activities specifically designed to prevent poor quality in products or services

**Appraisal costs:** The costs associated with measuring, evaluating or auditing products or services to assure conformance to quality standards and performance requirements

**Internal failure costs:** These are incurred when an error is detected in a product prior to shipment.

**External failure costs:** These are associated with defects found after the product has been shipped to the customer.

Of all the four aspects of the cost of quality, the external failure costs can be considered expensive. The prevention costs, appraisal costs, and internal failure costs are the costs that occur during the product development. They will be paid by the customer. But external failure costs are associated with defects found after the product has been shipped to the customer. So these costs must be bared completely by the developing side management.

External failure costs include costs like, complaint resolution, product return and replacement, labor costs associated with warranty work etc. A product replacement is not an easy issue. It involves much work and investment. The product should be revised and sometimes, it should be redesigned. Hence the external failure costs can be considere as the most expensive costs.

Problem:

Do a Web search and find three other examples of “risks” to the public that can be directly traced to poor software quality. Consider beginning your search at **http://catless.ncl.ac.uk/risks**.

Answer:

Low-quality software increases risks for both the developer and the end user. The downside of poorly designed and implemented applications does not always stop with dollars and time; they incur a great loss to people.

The examples of the risks to the public due to poor software quality are:

**Example 1:**

call centers dealing with emergency ambulance calls use software to automate the prioritization of calls. Once, a change was requested to downgrade the severity of incidents involving a fall of 10ft or more. The change was literally implemented with the consequence that all incidents involving a fall were downgraded, irrespective of the severity of other symptoms. The error came into light when a woman who had fallen 12ft, was unconscious and had breathing difficulties died after being left to wait because priority was given to a drunk who had collapsed on the street.

**Example 2:**

Due to an unforeseen side-effect of a software upgrade over the weekend in a bank, they had a problem with their online banking service. It happened that customers were able to access the details of other customers.

**Example 3:**

For a long time, metro train had no automatic door opening mechanism, so that anything trapped in a closing door stayed trapped. Once, a person’s leg became trapped in a train door as it left the station.

Problem:

Are *quality* and *security* the same thing? Explain.

Answer:

4633-14-11P SA: 9420

SR: 6376

**No**, quality and security are not the same things, but software security relates entirely and completely to quality. Software that does not exhibit high quality is easier to hack, and as a consequence, low-quality software can indirectl increase the security risk with all of its attendant costs and problems. So, to build a secure system, you must focus on quality.

Problem:

Explain why it is that many of us continue to live by Meskimen’s law. What is it about the software business that causes this?

Answer:

**Meskimen’s Law:** There’s never time to do it right, but always time to do it over.

Many of us continue to live by Meskimen’s law because of the high competition and demand for a product in the market. As there are tough competitors in producing software product, taking much time in developing high quality software, then it will take so long time to complete and it will be so expensive. And also, they will be out of business. Either ignores the market window, or simply exhausts all the resources.

So, people in industry try to produce an initial product which is good enough and not to be rejected anyway. Also in way that it does not involve so much perfectionism, and so much work that it would take too long or too much cost to complete. Hence these business requirements make the people to follow the Meskimen’ s law.

# Solution: Chapter 19: QUALITY CONCEPTS

19.1 Make list of your most important personal goals for college education and the life experiences the accompany it You would then list the detailed question that need to be answered to see how and how well a particular college met each of your goals

19.2 Answers will vary

19.3 It’ s certainly possible but not very likely. For example, it is doubtful that people spent a lot of time designing a classroom blackboard, yet the blackboard has been (and remains) and effective teaching took in many mathematics classrooms.

19.4 Answers will vary

19.5 Quality metrics are time invariant. If you build software that meets quality factors today, it will almost certainly continue to exhibit quality in years to come.

19.6 Answer will vary

19.7 The software dilemma which is trying to decide whether it is more cost effective to expend the resources neede to do something right the first time or spend your resources redoing something that the customer if not happy with.

19.8 Good enough software is a lot like student course project software. The software meets the customer’ s needs most important needs, but is not very robust when it comes to handling errors or unusual processing situations.

19.9 The external failure costs are likely to be the most expensive of the fours costs of quality. External failures are identified very late in the software life cycle. Most software engineering data suggests that the later an error is discovered the more expensive it is to correct.

19.10 Answers will vary

19.11Quality and security are not the same thing. Low quality software applications are often found to contain security weaknesses. Likewise it is possible to have highly secure software which is extremely difficult use (e.g. very time consuming authentication process). It is also possible to have software that is very easy to use, with poor securit (e.g. no logon procedure).

19.12 Project managers are notoriously bad at estimating time and costs. Customers frequently make last minute changes and additions to project requirements. This means that developers are always short of time and often find deadlines arriving faster than expected. It is easier to deliver visible functionality and hope the customer is happy tha to work on tasks the improve quality when the customer does not see the immediate benefits. Focusing on the forme is likely to get the next program increment to the customer on time, doing the latter may make cause the schedule t slip (even though not having to redo things would take less time in the long run).

mike6606
粉丝 - 0 关注 - 5

0 0

+加关注

« 上一篇： 软件工程 实践者的研究方法 第18章答案
» 下一篇： 软件工程 实践者的研究方法 第20章答案

posted @ 2021-01-21 15:33 mike6606 阅读(395) 评论(0) 编辑 收藏 举

刷新评论 刷新页面 返回顶

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

**编辑推荐：**
· 现代图片性能优化 - 图片资源的容错及可访问性处理
· 浅谈：服务架构进化论
· 我又和 redis 超时杠上了
· 巧用 CSS 变量，制作高级感拉满的网格动画
· 记录一次锁的优化

**阅读排行：**
· 使用 Vue 3 时应避免的 10 个错误
· ASP.NET Core Web API 接口限流
· 【故障公告】cc攻击又来了，雪上加霜的三月
· 现代图片性能优化及体验优化指南 - 图片资源的容错及可访问性处理
· 如何在 Net6.0 中对 WebAPI 进行 JWT 认证和授权