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 Research Methods for Practitioners Chapter 1 Answers

1.1 Problem:
Provide at least five additional examples of how the law of unintended consequences applies to computer software.

Answer:
The law of unintended consequences:

The law states that "while observing the problem in a deep way, it seems to be clear and a solution also seems to sure-fire. But at the same time the solution also creates another problem". Examples of the law of unintended consequences:

1. Boy and girl internet friend ship:

Instant messengers were developed as a solution for fast, real time communication.

The problem that rose with this type of software is that - strangers can communicate

with each other.

2. Way of chatting in mobile phone:

Mobile phone communication technology helped in erasing communication barriers.

But the same technology showed a path to anti-social elements in creating havocs.

3. War (Web – apps span a wide array of applications):

Computers are used in the every field, which was a change the strategies of the user. One of the examples is Digital automobile dashboard that shows an image in high quality resolution.

• This type of image was harmful to create an impact on the social network users.

• It can be little more than a set of lined hypertext files that present information using text and limited graphics.

• However, as e –commerce and B2B applications grow in importance, web apps are evolving in sophisticated computing environments.

4. Enjoying the videos in piracy DVD/media:

The advent of laser disc technology helped to store the data easily at low costs.

Due to the low cost, the audio and video material is getting pirated.

5. Missile tracking system:

It is a great achievement in embedded systems area. The Missile tracking system used to know about the paths of a missile.

• Actually it is a software system developed with an intension of finding out the path or to track the path generated o motion by a missile.

• But this has become curse. This technology is now being widely used for darker side rather than for a right purpose

1.2 Problem:
Provide a number of examples (both positive and negative) that indicate the impact of software on our society.

time information. It transforms personal data so that data can be more useful in a local context; it manages business information to enhance competitiveness ; it provides gateway to world wide information networks and provides the means for acquiring information in all of its forms. Importance of software in our society:

Computer software belongs in the abstract domain, defining its role in modern society in sometimes tricky. The computer and its peripherals are useless without a wide array of programs to make them useful.

Considering the immense presence of software in our everyday lives, industry and government, it can definitely be classed in an inexhaustible technological resource which supports users as they manipulate data, gathering and saving it in databases or files. When an organization uses software dedicated to its field of operations, it has the eventual tools to perform in a constantly changing environment. Software drawbacks:

Software engineering was spurred by the so called software crisis and which identified many of the problems of software development. Many software projects ran over budget and schedule. Some projects caused property damage. Some projects caused loss of life software crisis was originally defined in terms of productivity but evolved t emphasize quality.

1.3 Problem:
Develop your own answers to the five questions asked at the beginning of Section 1.1. Discuss them with your fellow students.

Answer:
Today, a huge software industry has become a dominant factor in the economies of the industrialized world. But yet questions that were asked are e

(1) Why does it take so long to get software finished:

Ans: Software development is an iterative process. User requirements are collected, through communication and the analysis is done on that and finally go for design. In between user can specify some more requirements and may nee some modifications. This takes development process to starting point. User may not be satisfied with the developed system, it has to undergo the refinement. There fore if take so long to get software finished. (2) Why are developmen costs so high?

Ans: Each and every software company has its own frame work. Any software is built on that framework. The framework itself costs so high and the maintenance of legacy software, people in the organization is not a smaller issue hence the development costs so high. (3) Why can' t we find all errors before we give the software to our customers?

Ans: *Customer point of view may be different from that of developers.

• All the possible test cases may not be available to developer

• Customer usage may differ with time

• Lack of proper communication.

• Excessive care taken by developers during development4) Why do we spend so much time and effort maintaining existing programs?

Ans: Developing a program is not alone a big issue but maintenance of the existing programs takes much effort.

Because we should have an idea of all the existing programs and the contexts in which they may be used. Rather thar developing the program from the scratch, it is very much useful to use the existing programs which will reduce the time and cost effectively. But we should have a great knowledge to identify the various situations where a particular program can fit. Therefore we spend so much time and effort maintaining existing programs.5) Why do we continue to have difficulty in measuring progress as software is being developed and maintained?

Ans: Software is not a physical quantity to measure it. Its progress is calculated in terms of complexity, time and efficiency. An efficient program developed for a time 'x hrs' may show bug at the last minute, which is a difficult scenario. Efficient program developed by an efficient developer may show an error during demonstration to the user.

Hence, still we have difficulty in measuring progress of software.

Many modern applications change frequently—before they are presented to the end user and then after the first version has been put into use. Suggest a few ways to build software to stop deterioration due to change.

Answer:
Build software to stop deterioration due to change

"Build the software is accommodating the changes rather than building it from the scratch".

It is easy to extend but difficult to modify.

Many modern applications change frequently. Before they are presented to the end _user, the first version is called as version.

To prevent low quality (deter) in change of developed software, developers must care about the following issues:

1. While changes are made in one part of the program, it has to be seen that no side-effect is created in another part of the program.

2. Software must be adapted to meet the needs of new computing environments or technology.

3. It also has to be kept in mind that the software should not depend on external devices or systems that are likely to change with time.

4. While doing the testing in an application, test cases and results are archived and available. So, software can be retested when changes are made.

5. First, time spending is needed to understand the requirements of the customer, what the customer wants.

6. Even a single minor requirement must not be neglected.

1.5 Problem:
Consider the seven software categories presented in Section 1.1.2. Do you think that the same approach to software engineering can be applied for each? Explain your answer.

Answer:
Software is engineered, not manufactured.

Nature of software:

The computer software is the product that is built by the software professionals, supported for a long term.

• It defined with the various programs and approaches presented in the form of architecture.

• Nowadays, software plays a major role because it affects every aspect of the human life in all the activities.

• In the world, it is placed as a most important technology to improve business, science, engineering, etc.,

• The importance of software is increasing and growing as software community developing the technologies continue to make the life easier, less expensive, faster, and high-quality programs are maintained.

• Software manages to play a dual role that, it acts as a product and as a vehicle (medium) to deliver a product; managing both personal and business information in any manner, also acts a medium for software threats.

• The role of computer software deals with the significant change over the years, to become a dominant factor in the industrialized world.

• Also, there are more concerns about the software being developed to examine the characteristics of the software's that are built by the human world.

Seven broad categories of computer software present continuing challenges for software engineers are shown below

1. System software

2. Application software

3. Engineering / scientific software

4. Embedded software

5. product – line software

6. Web – application

7. Artificial intelligence software.

Software Engineering approach:

Software engineering is an engineering discipline which involves all the aspects of software production from the starting stage to the final stage by maintaining the specifications of the system.

• It allows the professionals to build the high-quality computer software which encompasses a process, a collection o methods and the tools of the array.

• It completes the profession with a discipline of work to build a complex system in a timely manner.

• It is the application of a systematic, quantifiable, disciplined approach to the development, operation, and maintenance of the software.

• This study approaches the application engineering of the software with a technology of layers.

• It defines all the forms to its application software' s which should be engineered.

• It ultimately leads to the continuous process improvement to approach the effectiveness of software applications to software engineering.

• It provides the techniques how-to build the software that includes all the stages of the software development with the descriptive techniques.

• These tools provide the full support of the software development to go through all the stages of the software to be engineered.If the same approach to the software engineering can be applicable?

The approach to software engineering may/may not be applied for each of the software categories because,

• Computer science relates to the specializations of theory and fundamentals.

• Software engineering relates to the specializations of developing and delivering the softwares in a practical way.

• Software engineering is not only writing programs, not only science but an art.

• To develop new programs, the same old programs can be extended by including the new features to the existing approach.

• Software engineering providing tools for the software development.

• Millions of software engineers worldwide are hard at work on projects in one or more of the above mentioned 7 categories.

• Nowadays, the software plays a major role in front of an economy in all the developed countries and all the systems are controllable.

• In some cases, new systems are being built, but in others, existing applications are being corrected, adapted and enhanced.

• So that, the expenditure is spent on the software development reflects the significant fraction of GNP in all the developed countries.

• The legacy behind the generation will ease the burden of future software engineers.

The different approach is needed for software engineering because,

• It may be applied in a similar fashion that the "new challenges" faced by the engineers will come down under each of the seven categories to deliver effects that cannot be predicted.

•However, the software engineers will need to develop the processes that are adaptable and agile with the corresponding changes in terms of technology and the business over the next decades.

• Many of the software programs are getting old, outdated and the upgraded versions needed.

• So, the software engineers must be well prepared by instantiating a process to accommodate dramatic changes capable of handling all the situations concerned in the future.

Solutions: Chapter 1: The Nature of Software

1.1 Classic examples include the use of "digital automobile dashboards" to impart a high tech, high quality images. Appliances that "think;" the broad array of consumer electronics; personal computers (today, differentiated more by their software function than the hardware), industrial instrumentation and machines. All e-commerce applications are differentiated by software.

1.2 This is a good problem for classroom discussion (time permitting). Rather than focusing on cliché' ridden (albeit important) issues of privacy, quality of life, etc., you might want to discuss "techno fright" and how software can help to exacerbate or remedy it. Another interesting possibility is to use Neumann's "Risks" column in SEN to key discussion. You might also consider new attempts at software-based 'cash' economies, new modes of interactive entertainment, virtual reality, e-commerce, etc.

1.3 It takes software so long to be finished, because of following reasons

1. Facilities are not available on line.
2. Development tools do not work as expected.
3. Customer insists on the new requirements, requiring redesign and rework.
4. Product depends on the government regulations that change unexpectedly.
5. Strict requirements for compatibility with existing system require more testing, design, and implementation then expected.
6. Requirements to operate under multiple operating systems take longer to satisfy than expected.
7. Software project risk management takes more time then expected.
8. Dependency on a technology that is still under development lengthens the schedule.

Development costs are high:

1. Unacceptably low quality requires more testing, design and implementation work to correct then expected.
2. Development of the wrong software functions requires redesign and implementation.
3. Development of the wrong user interface results in redesign and implementation.
4. Development of extra software functions that are not required extends the schedule.

We can' t find errors before we give the software to our customer because of the following reasons;

1. Product depends on government regulation, which changes unexpectedly.
2. Product depends on draft technical standards, which change unexpectedly.
3. New development personnel sometimes are added late in the project.
4. Conflicts within teams sometimes results in poor communication and hence poor design
5. Sabotage by project management results in efficient scheduling and ineffective planning.
6. Sometimes the furnished components are poor quality resulting in extra testing, design and integration work and in extra customer –relationship management.

We continue to have difficulty in measuring progress as software is

developed as;

a) Sometimes the purpose of the project is not clear.

d) We need to review our work continuously.

e) A time check has to be maintained.

f) Project team has to be thorough and organized throughout the process.

1.4 Many modern applications change frequently before they are presented to the end user and then after the first versions have been used. The few ways to build software to stop deterioration due to change would be:

Gather the required information.

Designer and customer define the overall objectives for the software.

Identify the known requirements.

After building a prototype the developer uses an existing program fragment, this will help the working program to complete quickly.

To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.

Documents should be developed in a timely manner, to do this documentation standards are defined and mechanisms are established.

Review works done up to a particular stage.

There should be a backup person for every critical team member.

Check whether the risk aversion steps are being properly applied or not.

Check whether the necessary information for future risk analysis is necessary to collect.

1. 5 The same approach to software engineering can be applied for each of the seven categories. Each of these "new challenges" will undoubtedly have effects (for business people, software engineers, and end-users) tha cannot be predicted today. However, software engineers can prepare by instantiating a process that is agile and adaptable enough to accommodate dramatic changes in technology and business rules that are sure to come over the next decade.

mike6606
粉丝 - 0 关注 - 5

« 上一篇： 杭电oj 1.3.1 1236 排名
» 下一篇： 软件工程 实践者的研究方法 第二章答案

posted @ 2021-01-20 13:47 mike6606 阅读(232) 评论(0) 编辑 收藏 举报

· 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