

SOFTWARE ENGINEERING CLASS

CSD 5.5

PRINCESS-HANNAH ACKON

ASSIGNMENT 1

Question 1:

Identify three examples of software failures in Ghana and three examples of software failures and their consequences in the world.

Answer:

SOFTWARE FAILURES IN GHANA:

1. Ghana Water Company and SOFTtribe:

In October 2015, the Ghana Water Company Limited (GWCL) decided to consolidate its billing system across the country. This was to ensure efficient management of the billing system by moving away from paper bills to e-billing using cloud technology and mobile applications. GWCL had also developed a dummy customer app to pilot the digitization of bills. The firm, SOFTtribe is run by Herman Chinery-Hesse, Kisseih-Tetteh Antonio and David Kwamena Bolton. According to multiple sources, GWCL offered all its intellectual property documents on utility billing system to the SOFTtribe to guide the indigenous company to work on GWCL billing database. It was established that GWCL's e-billing system was fraught with a lot of disparities and complaints from customers, staff and the early users of the software. There are a lot of people who say they didn't consume that much but were given so much bill. When customers begin to complain about the measurement of their consumption and that they are not satisfied, demand is going to slow down, willingness to pay is also going to slow down. It was established that the e-billing solution had some undefined scope and that it was still under development.

2. UNIPASS SOFTWARE SAGA:

The Mass Action Committee had called for the suspension of the UNIPASS contract because they believe it is not an end-to-end system as they claim it must be. The Government of Ghana as part of its effort to maximize revenue from international trade thought of the idea of having a single window system at the country's entry point. The government has over the years engaged the services of entities like GCNet, WESTBLUE, SML, and Ghana Link among others. These

entities performed various functions within the scope of the single window system. Upon assumption to the highest office in the country, the reigning party awarded the UNIPASS contract to Ghana Link to replace what was being operated by GCNet/WESTBLUE. According to this committee, the UNIPASS software was far from being end-to-end because it did not include a manifest platform. The manifest module constitutes the basic and first Customs Declaration Entry Point and if this Manifest module is not present, it is impossible to process any Declaration.

3. ECG PREPAID METER FAULTS:

Some customers of the Electricity Company of Ghana (ECG) within Accra West area whose meters were replaced with the new meters have expressed frustrations over their inability to recharge their prepayment meter with the smart cash card after their credits were exhausted.

Most of the residents Ghana News Agency chanced upon at the ECG Station at Mile 7, said they have been sleeping without electricity power for days. They also expressed their frustrations over the fact that officials asked them to "go and come" without any fruitful results. Officials at the ECG Mile 7 Sub-station explained to the Ghana News Agency that the problem was due to migration unto new IT software which was developed when the new meters were introduced.

SOFTWARE FAILURES AROUND THE WORLD

1. FACEBOOK'S APP OUTAGE:

In the summer of 2019 Facebook users stopped being able to view or load images from the newsfeed. But it wasn't just Facebook's issue. Other of its apps had troubles as well. Instagram, WhatsApp and Messenger had an inconvenient outage. Its users couldn't send messages, media files or view "instastories". The malfunction angered a lot of users and made the top of complaints on the rival platform, Twitter. Facebook announced that everything was fixed and happened due to an accident during "routine maintenance". But these social media platforms are so popular that these kinds of things can affect a lot of people and cost the company a bad representation and ads refunds.

2. CPU'S FLAW:

Google had shocked everyone in 2018 revealing CPUs vulnerabilities. Meltdown and Spectre, which can access private and sensitive data of other programs. They influence Intel, AMD and ARM chips. This hardware flaw makes stealing data, passwords and keys a piece of cake. Furthermore, there are new versions of these vulnerabilities, like the one of Spectre called SWAPGSAttack. This tricky thing provides an opportunity to monitor computers and take information leaving no trace. And computers (workstations and laptops) aren't the only devices that can be affected. Smartphones, tablets, servers and others can also take a hit. Microsoft worked extremely hard to create software solutions to these hardware problems. And it had some success. Now there are certain patches that help to secure your data against the initial flaws. You just have to update your systems and they will take effect. The patches might help with protecting your information, but they might also influence the device's performance in a negative way.

3. BRITISH AIRWAYS GLITCH:

When the busiest month for airways came along, computer system completely went down. On the 7th of August 2019 over hundred flights of British Airways were cancelled and near to three hundred delayed. Thousands of passengers had to stay behind and wait crazy long hours in the packed airports. The check-in procedures had to be switched to manual which made the queues very long. The pattern of software failures over the last couple of years suggests poor computer management and calls for concern. Investors are already pretty worked up, since the financial risks with such issues are too high.

ASSIGNMENT 2

Question 1

The myths noted above are fading as the years pass, but others are taking their place. Identify two new myths under any of the categories (customer, developer and management). Explain each myth as detailed as possible.

Answer

Pressman, (1997) describes a number of common beliefs that software managers, customers and developers believe falsely. These are misleading and have caused serious problems. These myths come from the point of view of the management, customer and from the developer's point of view. Myths have a number of attributes that cause serious problems on software. There are three types of myths. They are management myth, customer myth and developer or practitioner myth.

A software myth from the customer's point of view is general statements of objectives is sufficient to begin writing a program. The reality is that a poor up-front definition is a major cause of failed software efforts. A formal detailed description of the information, domain, function, behavior, performance, interfaces, design, constraints and validation criteria are essential. This is because they can be determined only after thorough communication between developer and customer.

They also believe that project requirements continually change and that it is easy because software is flexible. The reality is even though it is true that software requirements change, the impact of change varies with the time at which it is introduced.

In the management point of view, they believe that there is a process and standard tools present are sufficient for developers to build the program. The fact is that the book of standards may exist but may not be in use. It may also be that the software practitioners may or may not be aware of this book of standards. Management tend to question whether or not the book reflects on modern engineering and how well it is streamlined to improve delivery time while focusing on quality of product.

They also think they have the latest machines after all they purchase the newest of computers. The reality is it takes much more than the latest model mainframes, workstation or personal computers to make high-quality software development

projects. Computer-aided software engineering (CASE) tools are more important than hardware for achieving quality products, yet the majority of software developers do not use them (Computer-aided software engineering (CASE) tools) effectively.

The practitioner's believe that they can always go back to resolve an issue when it arises. This is because when they are coding they may feel reluctant to return to the line that has an error. This is an act of negligence that can cause loss of properties and even life. Also, they believe that once the program is written and running then they have completed their project. The reality is the sooner you begin writing the code, the longer it will take to get it done. Industry data shows that between 60 to 80 percent of all effort expended on software will be expended after it is delivered to the customer for the first time.

Another myth is until a program is running, there is no way of accessing its quality. The reality is one of the most effective software quality assurance mechanisms can be applied from the inception of the project. This is known as the formal technical review. Software reviews are a quality filter that has been found to be more effective than testing certain classed of software defects.

ASSIGNMENT III

Question:

Write about two other process models which were not discussed in class, taking into consideration their advantages and disadvantages.

ANSWER:

The software development life cycle is a series of phases that provide a common understanding of the software building process. There are various types of the software development life cycle. They are the waterfall model, V-Shaped model, evolutionary model, spiral method, Iterative and incremental method and the agile development model. The focus on this write up will be on the v-shaped model and the agile method including their advantages and disadvantages.

The v-shaped model is an extension of the waterfall model. Instead of moving down in a linear form, the process steps are bent upwards after the implementation and coding phase to form a typical V-shape. In this model, the software requirements, the necessary technologies and tools are clearly defined and known.

The advantages of the v-shaped model are that it is simple to use, each phase has specific deliverables, there is a higher chance of success over the waterfall model due to the development of test plans early on during the life cycle and also verification and validation of the product is done in the early stages of product development. The disadvantages include, the fact that it is costly and requires more time in addition to a detailed plan. The v-shaped model does not provide a clear path for problems found during testing phases. Also, the software is developed during the implementation phase so that there will be no production of software prototypes.

The second model is the agile development model. This model is based on iterative and incremental development model where requirements and solutions evolve through collaboration between cross-functional teams. This model can be used with any type of the project but it needs more engagement from the customer and to be interactive. It can also be used when a customer needs to have some functional requirement ready within the shortest possible time and the requirements are not clear enough. This will enable more valuable and workable piece for software early which also increase the customer satisfaction. Its advantages include; decrease in the time required to avail some system features, there is face to face communication and continuous inputs from customer representative which leaves no room for guesswork. Finally, the end result is the high-quality software in the least possible time duration and a satisfied customer. Some of its disadvantages include; the ability and collaboration of the customer to express user needs, documentation is done at later stages, reduction in the usability of components and also it requires special skills for the team to perform some tasks.

ASSIGNMENT 4

Question 1

Explain the 5 w's and 1 h in software engineering

ANSWER:

The 5 w's and 1 h stands for the What, Why, When, Where, Who and How. What describes the initial thought of process in understanding the basics of the issue, project or scenario at hand. It is about the cognitive mapping of the scope of the issue, problem or scenario for better understanding of the fundamental problem.

Questions like, “What is the technology we are using for software development? What expertise is available to the team to help them understand the compatibility issue?” are asked.

Why entails clarifying why the issue, problem or project at hand occurred. It aims at identifying the triggers and rationalizes the occurrence of an issue or problem. Questions like “Why was this problem not identified at the start of the project? Why were the project team involved in the project not able to detect the problem?”.

When is about the time-stamping the occurrence of an issue or problem. It helps to understand the time of occurrence which could help sequence and mitigate triggers and the impact of an issue or problem. Questions like “When was the first problem detected? When were the compatibility issues mapped and discussed?” are asked.

Where is the element used to pinpoint the location or place of occurrence and hence could be helpful in identifying the people and other things present at the location which may have contributed the occurrence of an issue or problem. Some questions asked include “Where is the client system located? Where are the quality assurance/ testers located?”.

Who is the element that identifies the people who may have direct or indirect involvement in contributing to the issue or problem. Some questions asked include “Who is responsible for ensuring technological compatibility within the project team? Who is responsible for problem detection and escalation?”

Finally, how is used to examine the sequence of things, its triggers and how the resultant problem or issue unfolded. Questions asked include “What was the sequence of events that led to detection of problem? How are compatibility tests performed?”.

ASSIGNMENT 5

HOME OWNER Use Case Description

Use Case ID:	U.C 1		
Use Case Name:	INSTALL HOME SECURITY SYSTEM		
Process Owner:	PRINCESS-HANNAH ACKON	Last Updated By:	PRINCESS-HANNAH ACKON
Date Created:	19-09-2020	Date Last Updated:	22-09-2020
Primary Actor	HOMEOWNER		
Description:	The home owner enters a password to allow all other instructions. The home owner enquires about the status of the security zone. The home owner enquires about the status of the sensors. The home owner presses the panic button in case of an emergency.		
Pre conditions:	The system has been programmed for a password to recognize various sensors around the premises		
Post conditions:	The home owner can monitor his household as well as intensify security.		
Performance Goal:	The system is set to monitor sensors when the home owner leaves or remains in the household.		
Basic Workflow:	1.Homeowner observes 2. Homeowner enters password 3. Homeowner selects “stay” or “away” 4. Homeowner observes read alarm light to indicate that SafeHome has been armed.		

Alternative Workflow:	<ol style="list-style-type: none"> 1. Control panel is not ready: homeowner checks all sensors to determine which are open; closes them. 2. Password is incorrect (control panel beeps once): homeowner re-enters correct password. 3. Password not recognized: monitoring and response subsystem must be contacted to reprogram password. 4. Stay is selected: control panel beeps twice and a stay light is lit; perimeter sensors are activated. 5. Away is selected: control panel beeps three times and an away light is lit; all sensors are activated.
Category:	Essential: must be implemented
Trigger:	The homeowner decides to “set” the system, i.e., to turn on the alarm functions.
Risks:	<ol style="list-style-type: none"> 1. Should there be a way to activate the system without the use of a password or with an abbreviated password? 2. Should the control panel display additional text messages? 3. How much time does the homeowner have to enter the password from the time the first key is pressed? 4. Is there a way to deactivate the system before it actually activates?
When Available:	First Increment
Channel To Actor:	Via control panel interface
Frequency Of Use:	Many times per day
Secondary Actor(s)	<ol style="list-style-type: none"> 1. Support Technicians 2. Sensors
Channels To Secondary Actors:	<ol style="list-style-type: none"> 1.Support Technician: phone line 2. Sensors: Hardwired and radio frequency interfaces

ASSIGNMENT 6

Question 1

What are the deliverables of each phase in the software development life cycle.

Answer:

The software development life cycle is a simplified representation of a software process. Each process model represents a process from a particular perspective and thus only provides partial information about the process. Deliverables are tangible or intangible goods or services produced as a result of a project that is intended to be delivered to a customer. The software development lifecycle includes the requirement phase, analysis phase, design phase, development phase, testing phase and the operation and maintenance phase. This could be a document, software product or a server upgrade. The following are the deliverables obtained from each stage of the software development life cycle process.

The requirement phase involves decisions about the foundations of the software, this tells the development team what needs to be done. Information is gathered from the clients, end-users and anyone who will be directly or indirectly involved in using the software. The document or deliverable obtained is the business requirement specification.

At the analysis phase, the business requirement specification obtained at the requirement phase is then analyzed and then approved by the clients. This is done through the software requirement specification document. The key people involved in this phase are the project manager, business analyst and senior members in the team. The deliverable at this phase is the software requirement specification.

The design phase there are two levels of design that goes on. One is the high level design. The high level design gives the architecture of the software product to be developed and this is done by the designers and senior developers. The other level of design is the low level design. This is done by the senior developers. It describes how each and every feature in the software should work and how every component should work. The outcome of this phase is a high level and low level documents.

At the development phase, the building of the software commences as well as writing the code for the product. The outcome of this phase is the source code document and the developed product.

The testing phase begins as soon as the software is ready. The developed software is then sent for testing. Once quality assurance is determined and it is also error free, it goes to the next phase which is implementation. The outcome of this phase is the quality product and the testing artifacts.

The final phase is the operation and maintenance phase. This is done by the operation and implementation engineers. Once when the clients start using the

developed system, the actual issues start showing up need to be solved from time to time. Maintenance should be done as per the service level agreement.