Debugging made easy!

interface

“If there are no bugs in the software program, why would any IT employer need a programmer?”

“ If you can ever find a bug-free/ dust-free house, why would anyone need a servant? ”

BUGS... FIXES!

Dear reader, most of the intelligent and successful people read their project related documentation, such as requirement documents, design documents etc., page by page. They do so in order to understand exact requirements and the business value the project might bring to the end-customer. To achieve this, they need to study the documents in detail rather than flipping through the pages and going to the end in a hurry.

Hence, we request you to read through this small booklet, page by page, till the end and cognize the information stored in it especially for you and not to miss-out on a potential eye-opener.

This booklet is mainly about various bugs in SOFTware/ HEARTware and their serious consequences. It also gives general guidelines on the prevention and correction of software bugs. Most importantly, this talks about the

solution to deal with the HEARTware bug, which is present in every human being with/without his/ her knowledge.

"Bug" is a very common word used in software industry. From trainee to CEO, almost everyone uses this word frequently in their daily work. In fact, as mentioned earlier, if there are no bugs, most of the IT jobs are at risk.

Does this sound interesting? If so, sit back, relax and read through this booklet. Hope you will find it useful.

For easy reading and understanding, this booklet is arranged into following sections:

- What are software bugs? Why do they exist? What are the consequences?
- Guidelines to prevent and eliminate the bugs
- What is HEARTware bug?
- Perfect/one-time solution (not patch!) for HEARTware bug

I. What are software bugs? Why do they exist? What are the consequences?

The Oxford dictionary states that the word "bug" was first coined by Thomas Edison on 11th March 1889. Mr. Edison was plagued with false signals with early designs of his new invention, the Phonograph. He thought an imaginary insect had placed itself inside the Phonograph, which troubled him for two nights. He named it a "bug" [I.]

Over the time, this word slowly made its way into every day IT vocabulary. In the context of computers, a 'bug' may be defined as ***"any error, flaw, failure or fault in a computer program or system that causes it to produce any incorrect or unexpected results".***

Some of the reasons that cause bugs are:

- Incorrect and/or inadequate under standing of the requirements
- Competency gaps and/or Technical skill gaps
- Inadequate or lack of proper quality processes or non-adherence to quality standards
- Gaps in communication/ collaboration

The bugs, as we know, cause serious consequences to customer, service provider and the project team. The consequences include dissatisfaction and negative business impact to the customer; SLA violations and escalations to

service provider and to the project team re-work and potential impact on performance appraisal.

The following case studies help us to understand how a simple bug can cause serious impact.

Case Study 1 ":

Haven't you heard the recent news about ISRO launching a record of 104 satellites in 'one go' successfully? This was a remarkable feat and yet another proud moment for our space community and the nation of India. We have

come a long way in our rocket/satellite launching programs. Do you know that the first launch of the PSLV program (PSLV-D1) was a big failure?

The rocket carried IRS-1E satellite but could not deploy it as the mission failed due to a software error in on-board guidance and control processor. According to The Failure Analysis Committee, a large disturbance occurred at the point of second stage separation and one of the retro rockets failed due to software error, a simple 'overflow' error in a control parameter, in on board guidance and control processor. What it means is that the control software in the mother console was designed to handle variations in a par¬ticular parameter, between, let us say, plus (+) or minus (-) 99.99. Now when that parameter crosses, say, -99.99 and reaches -100.00, the seven characters in ¬100.00 could not be recognized and so the software ignores the bit representing the '-' (minus) sign. The result was that in the flight a control command geared to correct a parameter of say -99.99 was suddenly changed by default to that required for + 100.00, while the system was actually suffering from a deviation of -100.00. Thus the control command from the computer instead of

correcting an error actually compounded it. A simple bug in handling the boundary cases of key parameters resulted in the collapse of a huge rocket.

Case Study 2:

A hard-working and smart professional with ten years of IT experience one day released a code that contained serious bugs into the Live/Prod environment which resulted in huge impact to the customers and their business. Though many

advised him to be careful because of the criticality of the project, he did not heed their warnings which ultimately costed him his job. It was indeed an agonizing experience and

emotional trauma for him and his family. They had to go through this because he did not give enough importance to debug and fix the bugs.

II. Guidelines to prevent and eliminate the bugs

Bug/Defect Prevention

As the old saying goes, "prevention is better than cure", it is always better to prevent bugs than to deal with them when they occur. Let' s look at few guidelines that can be very useful in preventing bugs:

- ***Inter-team and intra-team communication and collaboration:*** Have a clear and open communication between all the project teams to understand the objectives, goals, schedules/ deadlines, SLAs, OLAs etc. And every one should work together towards the success of the project. Many issues can be prevented when the project teams are tightly-coupled and have regular and open discussions.

- ***Adhere to processes and guidelines:*** Quality processes are developed to avoid human errors. Make sure every team member understands and follows all the processes related to the work. The list of processes might include requirements development process, design process, coding/ development process or testing and review process. Using a well-developed and detailed checklist ensures that the team is following proper processes.

- ***Develop a strong self-review and peer-review processes:*** "Review process" is one of the key processes to prevent a lot of bugs. Ensure the team has a strong self-review and peer-review processes. Every 'work item' must be peer-reviewed and approved in order to prevent the casual errors/ bugs.

- ***Prevent bugs in every phase of SDLC (software development life cycle):*** The early we catch the bug, the better for us.

Early detection helps to prevent much bigger re-work and loss. Impact of a bug in "requirements" is much higher than the impact of a "coding" bug. So, prevent the bugs as much as possible in the early phases of SDLC.

- ***When you are in doubt, ask; don't assume:*** Many bugs are caused because the project teams "assume" certain things during SDLC which are not in line with the requirements of the customer. When the teams openly ask and clarify all their questions/ queries/ doubts, they can prevent majority of the bugs.

Bug/Defect detection and correction

Even though we take every step to prevent the bugs, in reality, unfortunately, we see bugs creeping into production. This may be because of undocumented scenarios, mishandling of corner-cases, wrong coding methods etc. To avoid the bugs going into production, we should

have strong bug/defect detection and correction processes.

- ***Defect Identification Mechanism:*** We should develop an Early Warning System (EWS), which will enable us to identify the issues/ defects early. It could be performing regular health checks of the systems or monitoring or predictive analytics. Communicating all the impacted parties, escalating at right time and keeping the stake holders informed of the issues help us avoid a great deal of heartburns.

- ***Good Quality control systems:*** Develop strong testing methodologies and test automation processes to detect bugs. They will ensure that the product works as expected, as per the specifications and requirements. Use production-equivalent test-data to unearth the hidden bugs. Make sure the product is tested for scalability, stability and performance as well.

- ***Right de-bugging tools:*** Once the issue is identified, right debugging tools/ mechanism will help to find out the root cause and help in correcting them. In this era of automation, software tools can be set-up to identify and fix the issues easily so that the overall quality can be improved.
- ***Defect Analysis:*** Understanding the nature of defects, trends in the defect identification and analysis on the root-causes will help to further fine-tune the defect detection processes to unearth and fix many of the hidden bugs.

Similar to the issues in software there are also issues in humans. While some of these issues can be prevented, some cannot be prevented but only can be fixed. Let's look at this aspect in following section.

III. What is HEARTware Bug?

Dear friend, we are living in an extremely unstable and disturbed world. Aren't we? We regularly come across and also read in the news several incidents like the following:

- *Broken family relationships:* divorce cases, property disputes causing family disturbances, dowry harassments etc.
- *Workplace politics:* due to jealousy, selfish motives and/or greed causing unfairness, job dissatisfaction etc.
- *Addictions to alcohol and/or drugs:* leading to health hazards, social stigma etc.

We also read about some extreme incidents like a husband killing his wife, a son killing his own mother, a three-year old girl brutally raped by an old man, a teacher molesting student etc.

The root-cause of all these incidents is 'the problem of human heart'. Someone rightly said - "**The heart is deceitful above all things and**

beyond cure. Who can understand it?"[1]. Outwardly a person may look decent, well-dressed and well-mannered, but his/her heart may be full of deceit and wrong motives. Most of these negative behaviors start as a casual and/or social habits but lead to dangerous situations later on. This is what we call a "HEARTware" bug. This is nothing but sin inside us that causes us to do these things. When we grow from childhood to adolescence, our behaviors change drastically. Most of the habits start because of growing-age-curiosity inside us. We start casually in a "try" and "find-out" method. Many of the unhealthy and immoral habits start during teens and twenties. But, unless they are controlled and curtailed, they will soon become a driving force that will control our life-style, relationships, finances and finally our destiny.

You have all seen a big truck. It is strong, it goes fast and it can haul a lot of things. Everyone is afraid of it when it passes with a lot of noise. But all of a sudden, the truck turns over. What happened? A small nail; yes, just a small nail, stops even the biggest truck. Sins are like that; they can breakdown your life.

Whether we acknowledge or not, the truth is that the "bug" is in every human being. **"If we claim we have no sin, we are only fooling ourselves and not living in the truth"**.[2] Please continue to read through to the final part of this booklet on how the bug can be fixed permanently.

IV. Perfect/one-time solution (not patch!) for HEARTware bug

A software system with a bug cannot fix by itself; it needs an update, upgrade or patch to fix the bug. Likewise all humans with HEARTware bug cannot fix themselves but need explicit and

external help. That "outside source" should be bugless (sinless) and one who knows in and out of the humans. He could be none other than Creator-God.

As we saw earlier sin, small or big, has serious consequences. **"The wages of sin is death/ destruction".**[3] **Sin can be removed only through forgiveness.** The Bible tells us **"without the shedding of blood there is no forgiveness".**[4] The Creator-God came down to earth in the form of human being in order to accomplish the task of "shedding the holy blood". He is none other than Jesus Christ. **He is holy, harmless, undefiled, and separate from sinners.**[5] He was born sinless and lived a sinless life. He went to the Cross, shed His precious and holy blood and offered Himself as a perfect sacrifice. He defeated and defied sin on the cross of Calvary. He rose again and was victorious over sin and death! He paid the price so that we can be forgiven and set free. He offers the free gift of forgiveness to everyone who comes to Him with a believing heart.

Now that you have read through the whole booklet, we request you to think and re-assess yourself. Are you struggling with some kind of "sin" bug? You really want to come out of it. But the more you try to come out, the more you are going into it. It is making you spend sleepless nights, wasting your money and straining your relationships. Isn't it? The good news is that there is a perfect solution in Jesus Christ who can set you free.

Three steps to have Jesus as your savior and receive "moksham" i.e., eternal heaven:

1. **Admit** you have messed up your life and need God to forgive you. If we confess our sins, God is faithful and just, and will

forgive us our sins and purify us from all unrighteousness.[6]

2. **Believe** that Jesus died on the cross for all your wrong doings and rose again from the death to give you victory. For God so loved the world that He gave His only begotten Son, that whoever believes in Him should not perish but have life even after death.[7]

3. **Confess** and choose Jesus Christ to be the in-charge of your life that will stay with you always to help you overcome every sin and temptation that might come on your way. If you confess with your mouth, 'Jesus is Lord,' and believe in your heart that God raised him from the dead, you will be saved.[8]

Thanks for reading!

We wish that you will have
a "bug-free" life!

General References

I Extracts from: http://theinstitute.ieee.org/tech-history/technology-history/did-you-know-edison-coined-the-term-bug

II https://en.wikipedia.org/wiki/PSLV-D1

http://reflections-shivanand.blogspot.in/2007/09/pslv-crash-failure-analysis.html

References from The Holy Bible

1 Jer. 17:9

2 1 John 1:8

3 Rom. 6:23

4 Heb. 9:22

5 Heb. 7:26

6 1 John 1:9

7 John 3:16

8 Rom. 10:9

Note: If you want any help/ counsel/ guidance, please contact Interface at interfacefellowship@gmail.com.

Disclaimer:

The ideas expressed in the booklet are the viewpoints of the author. Freedom of speech/expression is the right to articulate one's opinions and ideas that includes any act of seeking, receiving and imparting information or ideas, regardless of the medium used. The Constitution of India provides the right of freedom, given in articles 19, 20, 21 and 22, with the view of guaranteeing individual rights that were considered vital. Freedom of religion in India is a fundamental right guaranteed by Article 25-28 of the Constitution of India according to which every citizen of India has a right to profess, practice and propagate his/her beliefs peacefully.

Notes:

Notes:

Notes:

"If debugging is the process of removing software bugs, then programming must be process of putting them in."

- Edsger Dijkstra

interface

mobile: 990-885-1414 E-mail: interfacefellowship@gmail.com
Website : www.interfacehyd.org Like us on Facebook (group)