Tech Tonics

TIMSCDR Research Journal

Volume 2 2014-2015

TECH TONICS: TIMSCDR Research Journal

PUBLISHER
Thakur Institute of Management Studies, Career Development & Research
Thakur Educational Campus,
Shyamnarayan Thakur Marg,
Thakur Village, Kandivli (E), Mumbai – 400 101
Telephone: 022 - 2884 0484/91, 022 - 67308301/02
Fax: 022 – 28852527
Email: timscdr@thakureducation.org, vinita.gaikwad@thakureducation.org
Website: www.timscdrmumbai.in

AIM

Research and development has been transforming computing paradigms and technology in multidimensional directions. Tech Tonics aims to inculcate research culture among post graduate students and make them aware of new innovational happenings in the field of Information Technology.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO-1 : To enable students to gain knowledge across all domains of Information Technology with in-depth understanding of their applications.

PEO-2 : To enable students to analyze problems and to design and develop software solutions using emerging tools and technologies.

PEO-3 : To enable students to continue Life-long learning, Research and Entrepreneurial pursuit in their chosen fields.

PEO-4 : To develop communication, teamwork, and leadership skills necessary to manage multidisciplinary projects and serve the society as responsible and ethical software professionals.

PROGRAM OUTCOMES (POs)

1. Apply domain specific knowledge of computing and mathematics for designing of software solutions for defined problems and requirements.
2. Understand and analyze a problem and suggest feasible solutions.
3. Design, evaluate, and develop effective solutions for complex computing problems to meet desired needs.
4. Design and conduct experiments and use research-based methods to investigate complex computing problems.
5. Use appropriate techniques and software tools for computing activities.
6. Understand and commit to professional norms, regulations and ethics.
7. Recognize the need for and have the ability to engage in independent learning for continual professional development.
8. Understand and apply project management principles, as a member or leader in multidisciplinary environments.
9. Effectively communicate technical information, both oral and written with range of audience.
10. Analyze societal, environmental, cultural and legal issues within local and global contexts when providing software solutions.
11. Work as a member or leader in diverse teams in multidisciplinary environments.
12. Use Innovation and Entrepreneurship for creation of value and wealth.

CONTENTS

SR. NO.	TITLE	AUTHORS	PAGE NO.
1.	Advanced Automobile Steering Wheel	Paritosh Jadhav	1
2.	Efficient Ways of Requirement Gathering	Sufiyan Ziya Harimesh Singh Ruel Cardozo Neha Chauhan Deep Singh Sanket Kumar Dave Tushar Damecha Ram Kumar Yadav	3
3.	Quality Assurance with Inbuilt Quality Standards	Narendar Flora Rutik Shah Abhishekh Shelar Harshal Sharma	12
4.	Security Features of Operating System of Cell Phone	Yogija Prabhu Arunima Bhol Lorna Lopes Peter Squeria Swati Verma Chintan Kinderkhedia	15
5.	Selection of Programming Language	Sandeep Kamat Ashil Katkoria Angha Baikar Ankur Kanoujiya Afzal Khan Mukesh Koli Jay Kotecha Krutika Hariharan	21
6.	Research Article on Software Sizing Techniques	Narayan Mehta, Anuradha Mishra Ruban Nadar ,Johny Johnson Nishit Mohanan Indal pal Neeraj Mourya Spandan Jain	25
7.	Cyber Crime and Underground Economy	Swati Verma, Peter Sequeira, Yogija Prabhu, Arunima Bhol, Lorna Lopes	29
8.	Malware Threats and Security	Sushant More Chintan Kinderkhedin Rahul Darji Shameka Paradkar Sana Shaik	33

Editorial

Tech Tonics – TIMSCDR Research Journal is a collection of scholarly research papers written by students of MCA (Masters in Computer Applications) course, comprising of research work in the domains of Information Technology and Applications.

Tech Tonics enhances the research interests amongst students at the Post Graduate level and inculcates amongst them the ability of applying the research knowledge either analytically or practically when interacting with the IT Industry. Also, students develop the ability to think and innovate emerging technologies in the dynamic field of Information Technology as they work with diverse research topics.

The Journal represents research work in various specializations in Information Technology like – OOPs, Computer Organization and Architecture, Software Engineering, Data Structures, Operating Systems, Computer Network, DBMS, Computer Graphics, Network Security, Software Project Management, E-Business, Embedded Systems, Human Computer Interfaces, Wireless and Mobile Technology, Distributed Computing, Cloud Computing, Cyber Security, Multimedia Technology, Information System Security and Audit, Bioinformatics, Software Quality Assurance, etc. Efforts have been made at various levels during the selection and publication process to ensure authenticity of research work. Information showcased in the research work is a compilation of Primary and Secondary sources of data. Students of TIMSCDR get opportunity for sound exposure to the field and relevance of standard research work through this academic exercise of performing research and representing the same through research papers.

Finally, this Research Journal is a humble effort to encourage the young and resourceful minds of the students to do research using latest techniques, and innovate and pen down emerging ideas in the field of Information Technology and Applications.

Dr. Vinita Gaikwad

Advanced Automobile Steering Wheel

Touch Screen Interface System that Simplifies Control

Guided By: Sudarshan Sirsat

Paritosh Jadhav

Abstract- **This research paper pertains to an idea that can possibly change our automobile driving experience by providing an advanced touch screen interface to the driver on the steering wheel, from where he can access the various control systems that are currently found on the dashboard or even on the steering wheel in the form of buttons. Through this paper, I suggest to simplify not only one's driving experience, but also provide him an environment of safety whereby he can maintain a steady focus on driving the vehicle and prevent himself from getting distracted towards the dashboard. This research work can be included in simplifying embedded systems design.**

Initially, I have delved into the current dashboard designs and how it can possibly cause, not only ergonomic inconvenience, but also dangers to the passengers in the automobile. I have also given references to studies alluding to these issues.

Later, I discuss, how embedding this technology can change one's driving experience for good.

Keywords- Steering, Touch-Screen, Safety, Embedded System(s), Interface.

I. INTRODUCTION

As a personal experience that I have encountered while driving a car, I have found it quite inconvenient a good number of times to change the temperature or the shuffling music. In addition, there can be important phone calls that you cannot avoid, either ways.

Certainly this is not the first time that an improvement has been suggested to make a driver's experience easy and safe.

Since the 1970s, automakers have been trying to make the car dashboard safer with advanced features such as air bags. However, it was only since the 90s that things started taking a turn and embedded systems started becoming a part of automobiles.

Although there have a good number of changes since the 90s till today, they haven't essentially taken care of the most common problems that any driver would face when he wants to change the controls on the dashboard.

Following are the two systems on a dashboard that a driver would usually interact with.

1) The car's music system.
2) Temperature control.

In addition to that, a driver may also need to adjust his seat according to his comfort. Though it's common that drivers would usually do it by parking their car on the side lanes, why not make it happen when the car is in motion with the help of a touch screen interface.

So essentially, even though an average car today has a steering wheel that is equipped with some essential features, why not make it better with the help of embedded technology?

Let us take a look what all you can get in today's steering wheels:

1) Music and temperature controls (as listed above)
2) Call answering.
3) Fuel level and average speed status.(Only in some vehicles).

These controls are in the form of buttons are present on both the sides of the steering wheel.

I suggest introducing all these controls in the form of a touch screen interface on side of the screen that can help the driver to get a grip on all the controls more efficiently.

II. WHY THIS CAN MAKE A DIFFERENCE

Let me take a step by step approach to help you understand how this would really help to help drivers drive safely.

A) *Music Control:* Although the modern steering wheels have buttons that can help one to navigate music from the playlist and also change the radio stations, one cannot see what music is next on the list. Suppose if the driver wants to play only a particular track on the list, he can prefer to see the list on the small touch screen interface on the side of the steering and just touch to select it.

There is absolutely no need to take a look on the side at the dashboard. This is a very important feature since it can help a driver to accomplish comfortable and most importantly a safe driving.

B) *Temperature Control:* You can find this on the side of the steering wheel, so here it is only being converted from a button to a touch screen interface.

C) *Seat Adjustment:* Those who have known to drive cars to longer distances must have known this. However, with an embedded touch screen, one need not pull over the vehicle and adjust the seat length or it's recline. The touch screen interface will help the driver to adjust the seat length or it's reclining, thus making him ergonomically comfortable without halting the vehicle.

However, this means that the car's electronic systems will have to be sufficed with even sophisticated embedded systems, which will drive the costs up.

A) *Indicators and Wiper Control:* Using indicators and wiper control can be tedious sometimes. But even this function can be included in the suggested touch screen interface.

B) *Texting and Calling:* Texting while driving is said to be one of the major reasons for road accidents throughout the world. However, this touch screen interface will be equipped with embedded software that is capable of converting human speech to text and will be linked to the user's phone in the vehicle via Bluetooth that will send the message to the intended receiver. However, this model will require a more complex software interaction between the embedded software and the smart phone's operating system.

C) *Navigation:* Given that it's a fairly bad idea to have your touch screen interface cluttered with information, we can still have another screen on the other side of the steering wheel for GPS navigation. However, to make this more practicable, a need to change the whole design of the steering wheel may arise.

III. SOME PERTINENT FINDINGS

According to a study in the United States pertaining to auto accidents, in the year 2009, 5574 people were killed and 44800 were injured due to distracted driving. This includes distraction due to the infotainment systems that are available on the dashboard [2].

Another study showed that nearly 15% of auto accidents are due to the driver using the cell phone either for texting or calling [3].

A U.K study reveals that 26% of auto accidents in the country are caused because of the driver's cell phone usage, mostly texting [1].

These findings are only a small chunk of the statistics on a whole. It is known through years of findings that cell phone usage by the driver and dashboard distractions are a major reason for accidents.

Apart from that, any functional buttons that are present on the dashboard can also cause these accidents.

The use of a touch screen provides the driver with an interactive interface that helps him to control these functions with the help of an embedded system.

IV. TOUCH SCREEN INTERFACE AS AN INTERMEDIATE STAGE IN DRIVING TECHNOLOGY

As an obvious suggestion one would make, would be that the whole steering wheel design should be changed and made totally touch based instead of mechanical. This is the future of automobile steering if at all there will be any manually driven vehicles around.

However, this is not going to happen all of the sudden as there will be eventual stages that lead to the above given design for steering wheels.

Therefore implementing this simple design in order to get the usually used controls in the hands of the driver through a touch screen interface is a good idea, provided that automakers design the car steering wheels in a way that would make more room for the touch screen(s) to be accommodated in the steering wheel.

V. COST CONSIDERATIONS

Having an embedded system that directly interacts with the driver can shoot up the costs of that certain car model, however, as it has been observed in case of previously high end technologies such as air bags, this kind of driver interface will become very common in almost all the cars and hence won't be a cost problem after a couple of years from its introduction.

As noted earlier, this suggestion is like an intermediate stage in the development of an advanced human-vehicle system, where the automobile will have even complex and sophisticated embedded systems.

Till then, this technology will help drivers to control most of the common functions found on the dashboard or in the form of a physical buttons on the steering wheel.
Having a touch screen will enable them to have many functions on their options as compared to having buttons on the steering wheels.

VI. ACKNOWLEDGMENT

This project is a part of the Semester I of M.C.A for *'Design That Simplifies Implementation'*, where I have chosen to take embedded system in automobile as the main topic and I am thankful to professor Sudarshan Sirsaat for suggesting me to take embedded systems as an alternative to software applications.

VII. REFERENCES

[1] NHTSA: National Highway Traffic Safety, "Auto accident report 2009".
[2] NSC: National Safety Council, U.K
[3] Consumer Reports. org

Efficient Ways of Requirement Gathering

Guided By: Sudarshan Sirsat

Sufyan Ziya, Tanveer, Harimesh Singh, Reuel Cardoza, Neha Cauhan, Deep Singh, Sanketkumar Dave, Tushar Dhamecha, Ramkumar Yadav

Keywords- **Requirement gathering technique, Document, Method, Group, Interview.**

I. INTRODUCTION

Techniques involving visualization of the requirements like storyboards, prototypes, scenarios are helpful when you have a business user who may not be worried about the ins and outs of technical solution or have long attention duration for legalizing the requirements with users to let the analyst drive his discovery efficiently than just reading a document with a prospective user. The requirement gathering techniques may differ from one project to another. Some requirement gathering techniques may prove highly beneficial for you in one project but may not be as productive in the other project or for some other company. Therefore the usefulness of a technique is determined by its need and the kind of advantages it offers in a particular project. There are requirement gathering techniques that you must be aware of in order to manage the projects in a better way and run your business successfully.

II. BRAINSTORMING

Brainstorming is a group or individual creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its member(s). The term was popularized by Alex Faickney Osborn in the 1953 book Applied Imagination. Osborn claimed that brainstorming was more effective than individuals working alone in generating ideas, although more recent research has questioned this conclusion. Today, the term is used as a catch all for all group ideation sessions.

Osborn claimed that two principles contribute to "ideative efficacy," these being:

A. Defer judgment,
B. Reach for quantity.

Following these two principles were his four general rules of brainstorming, established with intention to: reduce social inhibitions among group members, stimulate idea generation, increase overall creativity of the group.

A. Focus on quantity:

This rule is a means of enhancing divergent production, aiming to facilitate problem solving through the maxim quantity breeds quality. The assumption is that the greater the number of ideas generated, the greater the chance of producing a radical and effective solution.

B. *Withhold criticism:*

In brainstorming, criticism of ideas generated should be put 'on hold'. Instead, participants should focus on extending or adding to ideas, reserving criticism for a later 'critical stage' of the process. By suspending judgment, participants will feel free to generate unusual ideas.

C. *Welcome unusual ideas:*

To get a good and long list of ideas, unusual ideas are welcomed. They can be generated by looking from new perspectives and suspending assumptions. These new ways of thinking may provide better solutions.

D. *Combine and improve ideas*:

Good ideas may be combined to form a single better good idea, as suggested by the slogan "1+1=3". It is believed to stimulate the building of ideas by a process of association.

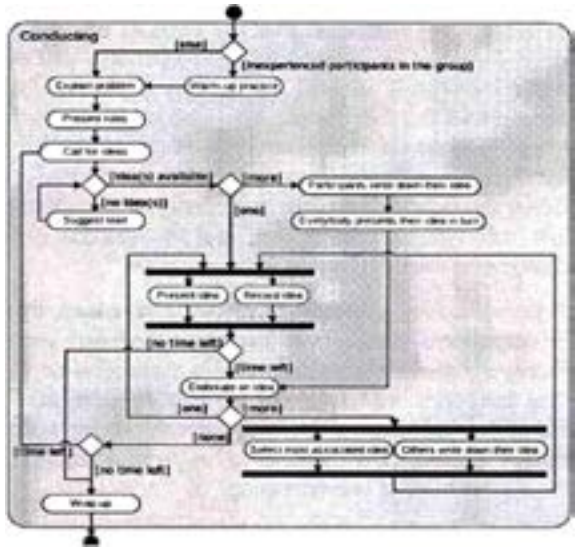


Fig. 1. Challenges to effective group brainstorming

A good deal of research refutes Osborn's claim that group brainstorming could generate more ideas than individuals working alone. For example, in a review of 22 studies of group brainstorming, Michael Diehl and Wolfgang Stroebe found that, overwhelmingly, groups brainstorming together produce fewer ideas than individuals working separately. Several factors can contribute to a loss of effectiveness in group brainstorming.

1) *Blocking:* Because only one participant may give an idea at any one time, other participants might forget the idea they were going to contribute or not share it because they see it as no longer important or relevant. Further, if we view brainstorming as a cognitive process in which "a participant generates ideas (generation process) and stores them in short-term memory (memorization process) and then eventually extracts some of them from its short-term memory to express them (output process)", then blocking is an even more critical challenge because it may also inhibit a person's train of thought in generating their own ideas and remembering them.

2) *Collaborative fixation:* Exchanging ideas in a group may reduce the number of domains that a group explores for additional ideas. Members may also conform their ideas to those of other members, decreasing the novelty or variety of ideas, even though the overall number of ideas might not decrease.

3) *Evaluation apprehension:* Evaluation apprehension was determined to occur only in instances of personal evaluation. If the assumption of collective assessment were in place, real-time judgment of ideas, ostensibly an induction of evaluation apprehension, failed to induce significant variance.

4) *Free-riding:* Individuals may feel that their ideas are less valuable when combined with the ideas of the group at large. Indeed, Diehl and Stroebe demonstrated that even when individuals worked alone, they produced fewer ideas if told that their output would be judged in a group with others than if told that their output would be judged individually. However, experimentation revealed free riding as only a marginal contributor to productivity loss, and type of session (i.e., real vs. nominal group) contributed much more.

5) *Personality characteristics:* Extraverts have been shown to outperform introverts in computer mediated groups. Extraverts also generated more unique and diverse ideas than introverts when additional methods were used to stimulate idea generation, such as completing a small related task before brainstorming, or being given a list of the classic rules of brainstorming.

6) *Social matching:* One phenomenon of group brainstorming is that participants will tend to alter their rate of productivity to match others in the group. This can lead to participants generating fewer ideas in a group setting than they would individually because they will decrease their own contributions if they perceive themselves to be more productive than the group average. On the other hand, the same phenomenon can also increase an individual's rate of production to meet the group average.

III. DOCUMENT ANALYSIS

Six questions must be addressed in document analysis:

- Which data are analysed?
- How are they defined?
- What is the population from which they are drawn?
- What is the context relative to which the data are analysed?
- What are the boundaries of the analysis?
- What is the target of the inferences?

a) *Personality characteristics:*

Extraverts have been shown to outperform introverts in computer mediated groups. Extraverts also generated more unique and diverse ideas than introverts when additional methods were used to stimulate idea generation, such as completing a small related task before brainstorming, or being given a list of the classic rules of brainstorming.

b) *Social matching:*

One phenomenon of group brainstorming is that participants will tend to alter their rate of productivity to match others in the group. This can lead to participants generating fewer ideas in a group setting than they would individually because they will decrease their own contributions if they perceive themselves to be more productive than the group average. On the other hand, the same phenomenon can also increase an individual's rate of production to meet the group average.

One more distinction is between the manifest contents (of communication) and its latent meaning. "Manifest" describes what (an author or speaker) definitely has written, while latent meaning describes what an author intended to say/write. Normally, content analysis can only be applied on manifest content; that is, the words, sentences, or texts themselves, rather than their meanings.

A further step in analysis is the distinction between dictionary-based (quantitative) approaches and qualitative approaches. Dictionary-based approaches set up a list of categories derived from the frequency list of words and control the distribution of words and their respective

categories over the texts. While methods in quantitative content analysis in this way transform observations of found categories into quantitative statistical data, the qualitative content analysis focuses more on the intentionality and its implications.

IV. FOCUS GROUP

A focus group is a form of qualitative research in which a group of people are asked about their perceptions, opinions, beliefs, and attitudes towards a product, service, concept, advertisement, idea, or packaging. Questions are asked in an interactive group setting where participants are free to talk with other group members. The first focus group was held in Ernest Dichter's house in a room he built above his garage. The first focus groups were created at the Bureau of Applied Social Research in the USA, by associate director, sociologist Robert K. Merton. The term itself was coined by psychologist and marketing expert Ernest Dichter.

A. Problems and criticism

A fundamental difficulty with focus groups (and other forms of qualitative research) is the issue of *observer dependency*: the results obtained are influenced by the researcher or his own reading of the group's discussion; raising questions of validity (see Experimenter's bias). Focus groups are "One shot case studies" especially if they are measuring a property-disposition relationship within the social sciences, unless they are repeated. Focus groups can create severe issues of external validity, especially the reactive effects of the testing arrangement. Other common (and related) criticisms involve groupthink and social desirability bias.

Another issue is with the setting itself. If the focus groups are held in a laboratory setting with a moderator who is a professor and the recording instrument is obtrusive, the participants may either hold back on their responses and/or try to answer the moderator's questions with answers the participants feel that the moderator wants to hear. Another issue with the focus group setting is the lack of anonymity. With all of the other participants, there cannot be any guarantee of confidentiality.

Douglas Rushkoff argues that focus groups are often useless, and frequently cause more trouble than they are intended to solve, with focus groups often aiming to please rather than offering their own opinions or evaluations, and with data often cherry picked to support a foregone conclusion. Rushkoff cites the disastrous introduction of New Coke in the 1980s as a vivid example of focus group analysis gone bad.

Jonathan Ive, Apple's senior vice president of industrial design, also said that Apple had found a good reason not to do focus groups: "They just ensure that you don't offend anyone, and produce bland inoffensive products."

B. Focus group data analysis

The analysis of focus group data presents both challenges and opportunities when compared to other types of qualitative data. Some authors have suggested that data should be analyzed in the same manner as interview data; while others have suggested that the unique features of focus group data - particularly the opportunity that it provides to observe interactions between group members - means that distinctive forms of analysis should be used. Data analysis can take place at the level of the individual or the group.

Focus group data provides the opportunity to analyze the strength with which an individual holds an opinion. If they are presented with opposing opinions or directly challenged, the individual may either modify their position or defend it. Bringing together all the comments that an individual makes in order can enable the researcher to determine whether their view changes in the course of discussion and, if so, further examination of the transcript may reveal which contributions by other focus group members brought about the change.

At the collective level, focus group data can sometimes reveal shared understandings or common views. However, there is a danger that a consensus can be assumed when not every person has spoken: the researcher will need to consider carefully whether the people who have not expressed a view can be assumed to agree with the majority, or whether they may simply be unwilling to voice their disagreement.

V. TIME CONSIDERITIONS

Many researchers underestimate the time required to complete a research project. The following form may be used as an initial checklist in developing time estimates. The best advice is to be generous with your time estimates. Things almost always take longer than we think they should.

This checklist contains two time estimates for each task. The first one (Hours) is your best estimate of the actual number of hours required to complete the task. The second one (Duration) is the amount of time that will pass until the task is completed. Sometimes these are the same and sometimes they are different. Most researchers and business-people have to divide their time among many projects. They simply cannot give all their time to any one project. For example, my estimate of goal clarification may be four hours, but other commitments allow me to spend only two hours a day on this study. My "hours" estimate is four hours, and my "duration" estimate is two days.

To arrive at your final time estimates, add the individual estimates. The hours estimate is used for budget planning and the duration estimate is used to develop a project time line.

SURVEY TIME PLANNING FORM

Task	Hours	Duration
Goal clarification		
Overall study design		
Selecting the sample		
Designing the questionnaire		
Write the cover letter		
Conduct pilot test		
Revise questionnaire (if necessary)		
Printing time questionnaire & cover letter		
Locating the sample (if necessary)		
Time in the mail & response time		
Attempts to get non-respondents		
Data entry and verification		
Coding open-ended responses		
Analyzing the data		
Preparing the report		
Printing & distribution of the report		

VI. COST CONSIDERATIONS

Both beginning and experienced researchers often underestimate the cost of doing questionnaire research. Some of the most common costs are:

1) Written Mail Survey Cost Planning Form

Task	Cost
Cover letter and questionnaire design & typing	
Purchasing mailing list cost (if necessary)	
Addressing mailing envelopes	
Following up on non-respondents	
Cover letter and survey printing costs	
Envelope costs (both ways + more)	
Postage costs (both ways + more)	
Incentives	
Data entry and verification	
Statistical analysis software	
Distribution of the final report	

2) Internet Survey Cost Planning Form

Task	Cost
Cover letter and questionnaire design & typing	
Purchasing email address list cost (if necessary)	

Bulk email sending software	
Survey hosting service	
HTML survey design software	
Incentives	
Statistical analysis software	
Distribution of the final report	

* Note: For Internet surveys, many software packages include the bulk email sending software, hosting service, HTML survey design software, and statistical analysis software bundled together.

VII. INTERFACE ANALYSIS

A. *Software interfaces*

A software interface may refer to a wide range of different types of interface at different "levels": an operating system may interface with pieces of hardware. Applications or programs running on the operating system may need to interact via streams, and in object oriented programs, objects within an application may need to interact via methods.

B. *Software interfaces in practice*

A key principle of design is to prohibit access to all resources by default, allowing access only through well-defined entry points, i.e. interfaces. Software interfaces provide access to computer resources (such as memory, CPU, storage, etc.) of the underlying computer system; direct access (i.e. not through well designed interfaces) to such resources by software can have major ramifications—sometimes disastrous ones—for functionality and stability.

The interface of a software module *A* is deliberately defined separately from the *implementation* of that module. The latter contains the actual code of the procedures and methods described in the interface, as well as other "private" variables, procedures, etc. Another software module *B*, for example the client to *A* that interacts with *A* is forced to do so *only* through the published interface. One practical advantage of this arrangement is that replacing the implementation of *A* by another implementation of the same interface should not cause *B* to fail—how *A* internally meets the requirements of the interface is not relevant to *B*, which is only concerned with the specifications of the interface.

Usually a method defined in an interface cannot be used directly; it must be implemented by non-abstract code that will run when it is actually invoked. An interface called "Stack" might define two methods: push() and pop(). It can be implemented in different ways, for example, FastStack and GenericStack—the first being fast, working with a stack of fixed size, and the second using a data structure that can be resized, but at the cost of somewhat lower speed.

An interface may define only a single method; for example, the Java language defines the interface Readable that has the single read() method. Various implementations are used for different purposes, including BufferedReader, FileReader, InputStreamReader, PipedReader, and StringReader. Marker interfaces like Serializable contain no methods at all.

C. *Programming to the interface*

The use of interfaces allows a programming style called programming to the interface. The idea behind this is to base programming logic on the interfaces of the objects used, rather than on internal implementation details. Programming to the interface reduces dependency on implementation specifics and makes code more reusable. It gives the programmer the ability to later change the behavior of the system by simply swapping the object used with another implementing the same interface.

Pushing this idea to the extreme, inversion of control leaves the context to inject the code with the specific implementations of the interface that will be used to perform the work.

VIII. INTERVIEW

When choosing to interview as a method for conducting qualitative research, it is important to be tactful and sensitive in your approach. Interviewer and researcher, Irving Seidman, devotes an entire chapter of his book, Interviewing as Qualitative Research, to the import of proper interviewing technique and interviewer etiquette. Some of the fundamentals of his technique are summarized below:

1) *Listening*: According to Seidman, this is both the hardest as well as the most important skill in interviewing. Furthermore, interviewers must be prepared to listen on three different levels: they must listen to what the participant is actually saying, they must listen to the "inner voice" or subtext of what the participant is communicating, and they must also listen to the process and flow of the interview so as to remain aware of how

tired or bored the participant is as well as logistics such as how much time has already passed and how many questions still remain. The listening skills required in an interview require more focus and attention to detail than what is typical in normal conversation. Therefore it is often helpful for interviewers to take notes while the participant responds to questions or to tape-record the interviews themselves to as to be able to more accurately transcribe them later.

2) *Ask questions:* While an interviewer generally enters each interview with a predetermined, standardized set of questions, it is important that they also ask follow-up questions throughout the process. Such questions might encourage a participant to elaborate upon something poignant that they've shared and are important in acquiring a more comprehensive understanding of the subject matter. Additionally, it is important that an interviewer ask clarifying questions when they are confused. If the narrative, details, or chronology of a participant's responses become unclear, it is often appropriate for the interviewer to ask them to re-explain these aspects of their story so as to keep their transcriptions accurate.

3) *Be respectful of boundaries:* Seidman explains this tactic as "Explore, don't probe," It is essential that while the participant is being interviewed they are being encouraged to explore their experiences in a manner that is sensitive and respectful. They should not be "probed" in such a way that makes them feel uncomfortable or like a specimen in lab. If too much time is spent dwelling on minute details or if too many follow-up questions are asked, it is possible that the participant will become defensive or unwilling to share. Thus, it is the interviewer's job to strike a balance between ambiguity and specificity in their question asking.

4) *Be wary of leading questions:* Leading questions are questions which suggest or imply an answer. While they are often asked innocently they run the risk of altering the validity of the responses obtained as they discourage participants from using their own language to express their sentiments. Thus it is preferable that interviewers ask open-ended questions instead. For example, instead of asking "Did the experience make you feel sad?" - which is leading in nature - it would be better to ask "How did the experience make you feel" - as this suggests no expectation.

5) *Don't interrupt*: Participants should feel comfortable and respected throughout the entire interview - thus interviewers should avoid interrupting participants whenever possible. While participants may digress in their responses and while the interviewer may lose interest in what they are saying at one point or another it is critical that they be tactful in their efforts to keep the participant on track and to return to the subject matter in question.

6) *Make the participant feel comfortable:* Interviewing proposes an unusual dynamic in that it often requires the participant to divulge personal or emotional information in the presence of a complete stranger. Thus, many interviewers find it helpful to ask the participant to address them as if they were "someone else," such as a close friend or family member. This is often an effective method for tuning into the aforementioned "inner voice" of the participant and breaking down the more presentational barriers of the guarded "outer voice" which often prevails.

IX. QUALITIES OF GOOD QUESTION

There are good and bad questions. The qualities of a good question are as follows:

1. *Evokes the truth:* Questions must be non-threatening. When a respondent is concerned about the consequences of answering a question in a particular manner, there is a good possibility that the answer will not be truthful. Anonymous questionnaires that contain no identifying information are more likely to produce honest responses than those identifying the respondent. If your questionnaire does contain sensitive items, be sure to clearly state your policy on confidentiality.

2. *Asks for an answer on only one dimension:* The purpose of a survey is to find out information. A question that asks for a response on more than one dimension will not provide the information you are seeking. For example, a researcher investigating a new food snack asks "Do you like the texture and flavour of the snack?" If a respondent answers "no", then the researcher will not know if the respondent dislikes the texture or the flavour, or both. Another questionnaire asks, "Were you satisfied with the quality of our food and service?" Again, if the respondent answers "no", there is no way to know whether the quality of the food, service, or both were unsatisfactory. A good question asks for only one "bit" of information.

3. *Can accommodate all possible answers:* Multiple choice items are the most popular type of survey questions because they are generally the easiest for a respondent to answer and the easiest to analyze. Asking a question that does not accommodate all possible responses can confuse and frustrate the respondent. For example, consider the question:

A. *What brand of computer do you own?*

- *IBM PC*
- *Apple*

Clearly, there are many problems with this question. What if the respondent doesn't own a microcomputer? What if he owns a different brand of computer? What if he owns both an IBM PC and an Apple? There are two ways to correct this kind of problem.

The first way is to make each response a separate dichotomous item on the questionnaire. For example:

- *Do you own an IBM PC? (Circle: Yes or No)*
- *Do you own an Apple computer? (Circle: Yes or No)*

Another way to correct the problem is to add the necessary response categories and allow multiple responses. This is the preferable method because it provides more information than the previous method.

What brand of computer do you own?
(Check all that apply)

- *Do not own a computer*

-IBM PC
-Apple
-other

4. *Has mutually exclusive options.* A good question leaves no ambiguity in the mind of the respondent. There should be only one correct or appropriate choice for the respondent to make. An obvious example is:

- *Where did you grow up?*

-country
-farm
-city

A person who grew up on a farm in the country would not know whether to select choice A or B. This question would not provide meaningful information. Worse than that, it could frustrate the respondent and the questionnaire might find its way to the trash.

5. *Produces variability of responses.* When a question produces no variability in responses, we are left with considerable uncertainty about why we asked the question and what we learned from the information. If a question does not produce variability in responses, it will not be possible to perform any statistical analyses on the item. For example:

What do you think about this report?

A. It's the worst report I've read
B. It's somewhere between the worst and best
C. It's the best report I've read

Since almost all responses would be choice B, very little information is learned. Design your questions so they are sensitive to differences between respondents. As another example:

- *Are you against drug abuse? (circle: Yes or No)*

Again, there would be very little variability in responses and we'd be left wondering why we asked the question in the first place.

6. *Follows comfortably from the previous question:* Writing a questionnaire is similar to writing anything else. Transitions between questions should be smooth. Grouping questions that are similar will make the questionnaire easier to complete, and the respondent will feel more comfortable. Questionnaires that jump from one unrelated topic to another feel disjointed and are not likely to produce high response rates.

7. *Does not presuppose a certain state of affairs:* Among the most subtle mistakes in questionnaire design are questions that make an unwarranted assumption. An example of this type of mistake is:

- *Are you satisfied with your current auto insurance? (Yes or No)*

This question will present a problem for someone who does not currently have auto insurance. Write your questions so they apply to everyone. This often means simply adding an additional response category.

- *Are you satisfied with your current auto insurance?*

-Yes
-No
-Don't have auto insurance

One of the most common mistaken assumptions is that the respondent knows the correct answer to the question. Industry surveys often contain very specific questions that the respondent may not know the answer to. For example:

- *What percent of your budget do you spend on direct mail advertising?*

Very few people would know the answer to this question without looking it up, and very few respondents will take the time and effort to look it up. If you ask a question similar to this, it is important to understand that the responses are rough estimates and there is a strong likelihood of error.

It is important to look at each question and decide if all respondents will be able to answer it. Be careful not to assume anything. For example, the following question assumes the respondent knows what Proposition 13 is about.

- *Are you in favour of Proposition 13?*

-Yes
- No
-Undecided

If there is any possibility that the respondent may not know the answer to your question, include a "don't know" response category.

8. *Does not imply a desired answer:* The wording of a question is extremely important. We are striving for objectivity in our surveys and, therefore, must be careful not to lead the respondent into giving the answer we would like to receive. Leading questions are usually easily spotted because they use negative phraseology. As examples:

- *Wouldn't you like to receive our free brochure?*
- *Don't you think the Congress is spending too much money?*

9. *Does not use emotionally loaded or vaguely defined words:* This is one of the areas overlooked by both beginners and experienced researchers. Quantifying adjectives (e.g., most, least, majority) are frequently used in questions. It is important to understand that these adjectives mean different things to different people.

10. *Does not use unfamiliar words or abbreviation:.* Remember who your audience is and write your questionnaire for them. Do not use uncommon words or compound sentences. Write short sentences. Abbreviations are okay if you are absolutely certain that every single respondent will understand their meanings. If there is any doubt at all, do not use the abbreviation. The following question might be okay if all the respondents are accountants, but it would not be a good question for the general public.

- *What was your AGI last year?*

11. *Is not dependent on responses to previous questions.* Branching in written questionnaires should be avoided. While branching can be used as an effective probing technique in telephone and face-to-face interviews, it should not be used in written questionnaires because it sometimes confuses respondents. An example of branching is:

1. Do you currently have a life insurance policy? (Yes or No) If no, go to question 3

2. How much is your annual life insurance premium?

These questions could easily be rewritten as one question that applies to everyone:

3. How much did you spend last year for life insurance? (write 0 if none)

12. Does not ask the respondent to order or rank a series of more than five items. Questions asking respondents to rank items by importance should be avoided. This becomes increasingly difficult as the number of items increases, and the answers become less reliable. This becomes especially problematic when asking respondents to assign a percentage to a series of items. In order to successfully complete this task, the respondent must mentally continue to re-adjust his answers until they total one hundred percent. Limiting the number of items to five will make it easier for the respondent to answer.

X. OBSERVATION

Observation is the active acquisition of information from a primary source. In living beings, observation employs the senses. In science, observation can also involve the recording of data via the use of instruments. The term may also refer to any data collected during the scientific activity

In some specific fields of science the results of observation differ depending on factors which are not important in everyday observation. These are usually illustrated with "paradoxes" in which an event appears different when observed from two different points of view, seeming to violate "common sense".

- *Relativity:* In relativistic physics which deals with velocities close to the speed of light, it is found that different observers may observe different values for the length, time rates, mass, and many other properties of an object, depending on the observer's velocity relative to the object. For example, in the twin paradox one twin goes on a trip near the speed of light and comes home younger than the twin who stayed at home. This is not a paradox: time passes at a slower rate when measured from a frame moving with respect to the object. In relativistic physics, an observation must always be qualified by specifying the state of motion of the observer, its reference frame.

- *Quantum mechanics:* In quantum mechanics, which deals with the behavior of very small objects, it is not possible to observe a system without changing the system, and the "observer" must be considered part of the system being observed. In isolation, quantum objects are represented by a wave function which often exists in a superposition or mixture of different states. However, when an observation is made to determine the actual location or state of the object, it always finds the object in a single state, not a "mixture". The interaction of the observation process appears to "collapse" the wave function into a single state. So any interaction between an isolated wave function and the external world that results in this wave function collapse is called an *observation* or *measurement*, whether or not it is part of a deliberate observation process.

XI. REQUIREMENT WORKSHOP

A. Customer Requirements

Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS). The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, answer the questions posed in the following listing:

- *Operational distribution or deployment*: Where will the system be used?
- *Mission profile or scenario*: How will the system accomplish its mission objective?
- *Performance and related parameters*: What are the critical system parameters to accomplish the mission?
- *Utilization environments*: How are the various system components to be used?
- *Effectiveness requirements*: How effective or efficient must the system be in performing its mission?
- *Operational life cycle*: How long will the system be in use by the user?
- *Environment*: What environments will the system be expected to operate in an effective manner?

1) *Architectural Requirements :*

Architectural requirements explain what has to be done by identifying the necessary systems architecture of a system.

2) Structural Requirements

Structural requirements explain what has to be done by identifying the necessary structure of a system.

3) Behavioral Requirements

Behavioral requirements explain what has to be done by identifying the necessary behavior of a system.

4) Functional Requirements

Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirements analysis will be used as the top level functions for functional analysis.

5) Non-functional Requirements

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors.

6) Core Functionality and Ancillary Functionality Requirements

Murali Chemuturi defined requirements into Core Functionality and Ancillary Functionality requirements. Core Functionality requirements are those without fulfilling which the product cannot be useful at all. Ancillary Functionality requirements are those that are supportive to Core Functionality. The product can continue to work even if some or all of the Ancillary Functionality requirements are fulfilled but with some side effects. Security, safety, user friendliness and so on are examples of Ancillary Functionality requirements.

7) Performance Requirements

The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements.

8) Design Requirements

The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals.

9) Derived Requirements

Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.

10) Allocated Requirements

A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items.

REFERENCES

[1] Requirements Engineering Elizabeth Hull, Ken Jackson, Jeremy Dick – 2010

[2] Software Testing: Principles and Practice Srinivasan Desikan, Gopalaswamy Ramesh – 2006

[3] Issues of Human Computer Interaction Anabela Sarmento - 2005

[4] Software Engineering with Reusable Components Johannes Sametinger – 1997

[5]www.statpac.com

[6]www.wikihow.com

[7] 40 Ways To Gather Information On The Internet and to Use It Efficiently http://www.corbinball.com/articles/art-info.html

Quality Assurance with Inbuild Quality Standards

Guided By: Sudarshan Sirsat

Narender Flora, Rutik Shah, Abhishek Shelar, Harshal Sharma

***Abstract*—This paper is trying to put its study ahead about how quality assurance can be obtained by the inbuilt quality standards within an application through code reusability and component sizing development methods.**

Keywords—Quality Assurance; ISO; quality standards.

I. INTRODUCTION

What Is Quality Assurance?

Quality Assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers.

QA is applied to physical products in pre-production to verify what will be made meets specifications and requirements, and during manufacturing production runs by validating lot samples meet specified quality controls.

QA is also applied to software to verify that features and functionality meet business objectives, and that code is relatively bug free prior to shipping or releasing new software products and versions.

Quality Assurance refers to administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled.

It is the systematic measurement, comparison with a standard, monitoring of processes and an associated feedback loop that confers error prevention. This can be contrasted with quality control, which is focused on process output.

Two principles included in Quality Assurance are: "Fit for purpose", the product should be suitable for the intended purpose; and "Right first time", mistakes should be eliminated.

QA includes management of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes.

Suitable quality is determined by product users, clients or customers, not by society in general. It is not related to cost, and adjectives or descriptors such as "high" and "poor" are not applicable.

For example, a low priced product may be viewed as having high quality because it is disposable, where another may be viewed as having poor quality because it is not disposable.

What is Quality?

The meaning of quality differs depending upon circumstances and perceptions.

For example, quality is a different concept when focusing on tangible products versus the perception of a quality service.

The meaning of quality is also time-based or situational.

Common Meanings of Quality:

A. Quality is fitness for use:

Quality means the product or service does what it is intended to do. Poor quality of a product or service cost users if it doesn't do what it is supposed to do.

B. Quality is meeting customer expectations:

Quality is satisfying the customer. The customer defines quality. The customer perceives the quality of a product or service.

C. Quality is exceeding the customer expectations:

Quality is the extent to which the customers or users believe the product or service surpasses their needs and expectations. Quality is delighting the customer.

D. Quality is superiority to competitors:

Quality is how a company's products and services compare to those of competitors or how they compare to those offered by the company in the past.

What Is Software Quality?

Structural quality is evaluated through the analysis of the software inner structure, its source code, at the unit level.

The technology level and the system level, which is in effect how its architecture adheres to sound principles of software architecture outlined in a paper on the topic by OMG.

In contrast, functional quality is typically enforced and measured through software testing

Software quality measurement quantifies to what extent a software or system rates along each of these five dimensions.

An aggregated measure of software quality can be computed through a qualitative or a quantitative scoring scheme or a mix of both and then a weighting system reflecting the priorities.

This view of software quality being positioned on a linear continuum is supplemented by the analysis of "critical programming errors" that under specific circumstances can lead to catastrophic outages or performance degradations that make a given system unsuitable for use regardless of rating based on aggregated measurements.

Such programming errors found at the system level represent up to 90% of production issues, whilst at the unit-level, even if far more numerous, programming errors account for less than 10% of production issues.

II. RESEARCH WORK

- Code Quality
- Ensure quality from the start.
- Quality is not something that can be easily added later.
- Problems that are too complex, too obscure, or are discovered too late in the product cycle are usually not fixed.
- Good code makes you feel good when you read it.
- In other words, code quality is related to something that happens behind a programmer's eyes, not some external metric.

It is subjective because different programmers have different expectations based both on the amount of experience they have and the tradition in which they were trained.

Proper Standard Training :Raising employees' skill levels to their maximum through proper training is a vital investment in every business.
Because the growth of an employee results in the growth of an entire organization.

From entry level to management level, training is designed to capitalize on employee strengths and develop their potential for excellence.
Training can modify the corporate culture, promote team cohesiveness, and improve communications between employees to create the optimum environment for productivity.

Employees also can learn how to prioritize tasks, become more effective leaders, and motivate employees.

Writing Clear Code:

- The overarching goal when writing code is to make it easy to read and to understand.
- Well-written programs are easier to debug, easier to maintain, and have fewer errors.
- Writing a program is a lot like writing an essay.
- When writing an essay, your message is more convincing when it is accompanied by proper grammar and punctuation.
- When writing computer programs, you should follow the same principle.
- It is even more important when programming since someone may be assigned to maintain and support your code for long periods of time.

You will appreciate the importance of good style when it is your task to understand and maintain someone else's code!
Coding.

- Keep programs and methods short and manageable.
- Use language-specific idioms.
- Use straightforward logic and flow-of-control.
- Avoid magic numbers (numbers other than -1, 0, 1, and 2); instead, give them meaningful symbolic names.

Internal audit done by PM/TM

Internal auditing is an independent, objective assurance and consulting activity designed to add value and improve an organization's operations.

It helps an organization accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes.

Internal auditing is a catalyst for improving an organization's governance, risk management and management controls by providing insight and recommendations based on analyses and assessments of data and business processes.

With commitment to integrity and accountability, internal auditing provides value to governing bodies and senior management as an objective source of independent advice.

Professionals called internal auditors are employed by organizations to perform the internal auditing activity.

III. GOAL OF STUDY

A. RELATED STANDARDS

The ISO 9000 family of quality management systems standards is designed to help organizations ensure that they meet the needs of customers and other stakeholders while meeting statutory and regulatory requirements related to a product.ISO 9000 deals with the fundamentals of quality management systems, including the eight management principles upon which the family of standards is based. ISO 9001 deals with the requirements that organizations wishing to meet the standard must fulfill.

Third-party certification bodies provide independent confirmation that organizations meet the requirements of ISO 9001. Over one million organizations worldwide are independently certified, making ISO 9001 one of the most widely used management tools in the world today. Despite widespread use, the ISO certification process has been criticized as being wasteful and not being useful for all organizations.

Overview of ISO 9001

- ISO (INTERNATIONAL ORGANISATION FOR STANDARDIZATION)
- Mainly addresses operational & organizational aspects like responsibilities, reporting etc.
- It is a set of guidelines for production process & is not directly concerned with the product itself.
- It says that, if proper process is followed for production then good quality products are bound to follow automatically.
- ISO 9001 → the international standard for the quality management of businesses.
- ISO 9001 → ISO's most well known standards.
- They are implemented by more than a million organizations in some 175 countries.
- ISO 9001 → for quality management

A. Quality management means:

what the organization does to ensure that its products or services →satisfy the customer's quality requirements and → confirm with any regulations applicable to those products or services.
It also means what the organization does to enhance customer satisfaction, and achieve continual improvement of its performance.
ISO 9001 → are generic standards
Generic means that the same standards can be applied:
to any organization, large or small, whatever its product or service in any sector of activity, whether it is a business enterprise, a public administration, or a government department.
ISO 9001 concern the way an organization goes about its work.
They are process standards.

IV. RESULT

ISO 9001 requirements

- Management responsibility
- Quality system
- Contract reviews
- Design control
- Document control
- Purchasing
- Purchaser supplied product
- product identification
- Process control
- Inspection and testing
- Inspection, measuring and test equipment
- Inspection and test status
- Control of nonconforming products (NC)
- Corrective action
- Handling
- Quality records
- Quality audits
- training

Online Research

How to get ISO 9000 Certification?

- Application:
- Application to the registrar.
- Pre-assessment :
- The registrar makes the rough assessment of the organization
- Document review and adequacy of audit:
- The registrar reviews the documents submitted and makes suggestion for possible improvement.
- Compliance audit: The registrar checks weather the suggestions made by him have been complied with by the org. or not
- Registration: the registrar awards the ISO 9000 certificate
- Continued surveillance: The registrar continues to monitor the organization, through periodically.

THE WAY FORWARD

- Benefits of implementing ISO 9001.
- It will motivate staff by defining their key roles and responsibilities.
- Cost savings can be made.
- Orders will met consistently on time and to correct specification.
- Wastage will be less.

V. CONCLUSION

Conclusion 1: Program managers are well aware of the risks factors for their programs and are managing them well, within the circumstances and resources available to them.

Conclusion 2: While the risk of errors is well-managed across all programs reviewed, some would benefit more than others from additional investment to further lower the risk of errors.

Conclusion 3: Human resources are the dominant risk factor.

Conclusion 4: The existence of a research and analysis capacity separate from the production operation is a key factor in assuring quality.

Conclusion *5:* There are numerous "best practices" in all programs that can usefully be shared.

Conclusion 6: All mission critical programs should have a strong and explicit research and analysis capacity, separate from the production operation, whose role is to challenge the data and to conduct research into the particular subject matter of the program. The most suitable form of this capacity may vary across programs.

Conclusion 7: The interdependency among the programs is an important factor in the quality of the results. Risks often increase when data flow across organizational boundaries and these flows should be explicitly managed.

Conclusion 8: The review of quality assurance practices was very useful and should become an ongoing program.

REFERENCES

[1] https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-rtl-sd-wp&S_PKG=ov13053&S_TACT=C25600FW&iio=BSWG&jm=-&cmp=C2560&ct=C25600FW&cr=google&cm=k&csr=Unbranded%7CSearch%7CUse+Service+Virtualization+to+Remove+Testing+Bottlenecks%7CROW%7Cov13053%7C1047&ccy=us&ck=software%20+quality%20+assurance&cs=b&cn=Software_quality_assurance-1047&mkwid=sLE8lhLNz-dc_40915066556_43246d30503

[2] en.wikipedia.org/wiki/Quality_assurance.

[3] http://asq.org/learn-about-quality/quality-assurance-quality-control/overview/overview.html.

Security Features of Operating Systems of Cell Phones

Guided By: Sudarshan Sirsat
Arunima Bhol, Lorna Lopes, Yogija Prabhu, Peter Sequeira, Swati Verma

I. INTRODUCTION

In the recent period, mobile operating systems are getting a tremendous growth. Analysts are upbeat further more potential developments in coming days. Today's markets, a wide variety of mobile phones are available in various brands with different operating systems. So the selection of an optimal and secured one is a confusing task. Present mobile operating systems are equally or more playing vital role than a computer operating system. It is covering most of the activities which were doing in a computer. We can see many of them are highly relay on mobile phones for their office activates like outlook, communicator etc. That indicates mobile operating systems are leveraging the high capability. Security is one of the key features of any mobile operating system. Many studies are keeping progressing in the same area. These studies are trying to provide a glance about the various mobile operating systems and its security measures which were expecting and available.

At present the usage of mobiles are very high, same way the expectations also. Many of the daily activities got replaced by mobile from personal computer, so the respective user expectations also increased. Below are the highly using functionalities through mobile.

A. Search for information
B. Access social networks
C. Access local information and services
D. Search Videos
E. Accessing WebPages
F. Entertainments
G. Shopping
H. Wi-Fi, Bluetooth and GPS connectivity
I. Travel information
J. Office activities e-mail, communicator etc.

II. STATISTICAL ANALYSIS

Many operating systems are available in the current market. As per latest statistics Google's Android and Apple's iOS are in front in the utilization. All operating systems are in race to bring new features and getting a leading role in the current market. Below diagram (Fig 2) and table (Table 2) give a better understanding.

Table 2. Market coverage of Mobile OS

Operating System	Percent
Android	37.19
iOS	27.18
Series 40	12.56
SymbianOS	7.98
Samsung	4.7
Unknown	4.28
BlackBerry OS	3.27
Windows Phone	1.18
Sony Ericsson	0.63
bada	0.41
Nintendo 3DS	0.16
Playstation	0.1
Linux	0.08
	0.07
Nintendo	0.04
Motorola	0.03
LG	
MeeGo	0.03
Windows	0.02
WAP	0.01

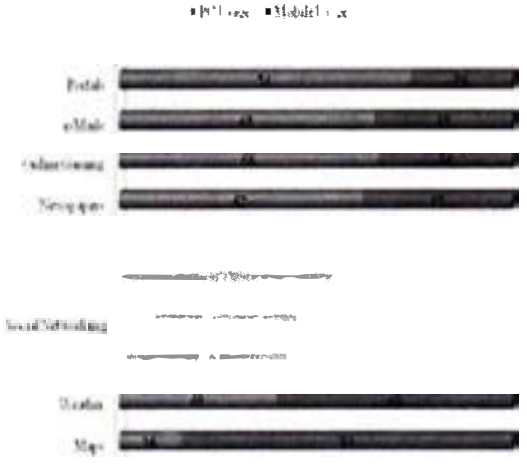
Current era each and every activity has directly or indirectly relaying the internet features. Recent days 40% of internet usage is going through mobile devices. That is disclosing the importance of mobile operating systems. In 2008 Mary Meeker an analyst was predicted as "Mobile to overtake fixed Internet access by 2014". Most of the surveys are confirming that words. Nowadays mobile devices are highly using than personal computers for social network sites, weather reports, location maps, hearing music etc. Below diagram (Fig 3) and table (Table 3) give a better understanding.

Table 3. PC Vs Mobile Internet Usage

Items	PC Usage %	Mobile Usage%
Total Internet	60	40
Maps	16	84
Weather	40	60
Music	43	57
Social Networking	45	55
Sports	54	46
Retail	62	38
Newspapers	62	38
Online Gaming	66	34
e-Mails	65	35
Portals	74	26

Table 4. Mobile Browser Usage Percentage

Browser Version	Percentage
Safari 6.0	39.80%
Android Browser 4.0	21.93%
Safari 5.1	5.09%
Safari 536	5.00%
Opera Mini 4.2	3.02%
Chrome 28.0	2.58%
BlackBerry	2.51%
Chrome 18.0	1.97%
Safari 5.0	1.96%
Opera Mini 7.1	1.89%
Safari 8536	1.80%
Opera Mini 4.1	1.08%
Microsoft Internet Explorer 9.0 Mobile	0.96%
Silk	0.85%



Quick Glance to Leading Mobile OS and Security Features Provided By Them:

A. Android:

Android operating system is the most popular in the current mobile platforms. This is designed by Google using Linux kernel. Every day more than 1 million new Android based devices are activated in worldwide. It is an open source platform, so many of mobile manufactures are customizing this and using as there key operating systems. Android use Linux kernel as its hardware abstraction layer between hardware and other software. This also provides a better memory management, process management, security options and network options. This is written in Java programming language and run in the Dalvik virtual machine. The key advantage of this operating system is anyone can customize this operating system.

With the same reason the innovative growth of this OS is in peak. Android applications have common structure, mainly in views, recourse management, notifications, data storage, services etc.

Safety and security: As Android OS is an open platform; the security mechanism is a key challenge. Here in the base structure (Fig 5) is providing security check in application level.

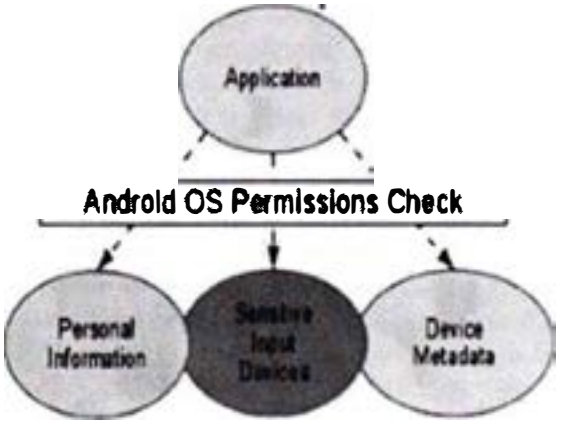


Fig 5: Base security structure

Securing an open platform requires robust security architecture and rigorous security programs. Android was designed with multi-layered security that provides the flexibility required for an open platform, while providing protection for all users of the platform. It was designed with device users in mind. Users are provided visibility into how applications work, and control over those applications. This design includes the expectation that attackers would attempt to perform common attacks, such as social engineering attacks to convince device users to install malware, and attacks on third-party applications on Android. Android was designed to both reduce the probability of these attacks and greatly limit the impact of the attack in the event it was successful. It seeks to be the most secure and usable operating system for mobile platforms by re-purposing traditional operating system security controls to:

- Protect user data
- Protect system resources (including the network)
- Provide application isolation
- Android provides these key security features:
- Robust security at the OS level through the Linux kernel
- Mandatory application sandbox for all applications
- Secure inter process communication
- Application signing.
- Application-defined and user-granted permissions

B. iOS:

iOS is one of the leading mobile operating system, which others trying to catch up. Related to the same many patent related cases are going in many places. This OS is designed by Apple followed by Mac operating system. In first glance the user friendly feature is the key for this operating system. HTML5 technology started rising in mid of 2011, iOS very well using this one. iOS considered the foundation of the iPhone. This is designed for the iPhone but now supports iPod touch, iPad and Apple TV Businesses around the world are choosing iOS devices for their enterprise-ready features and powerful security. iOS works with Microsoft Exchange and standards-based servers to deliver over-the-air push email, calendar, and contacts. It protects your data by encrypting information in three separate areas: in transmission, at rest on the device, and when backed up to iTunes. You can securely access private corporate networks through industry-standard VPN protocols. And companies can easily deploy iPhone across an enterprise using configuration profiles.

Safety and security: iOS provides built-in security from the moment you turn on your device. Low- level hardware and firmware features are designed to protect against malware and viruses, while high- level OS features help to secure access to personal information and corporate data. To guard your privacy, apps requesting location information or data from Calendar, Contacts, Reminders, ard Photos must first get your permission. You can set a pass code lock to prevent unauthorized access to your device and configure it to delete all your data after too many unsuccessful pass code attempts. This pass code also automatically encrypts and protects your stored email as well as allows third- party apps to encrypt their stored data. iOS supports encrypted network communication that apps can use to protect your sensitive information during transmission. And, in case your device is lost or stolen, Find My iPhone allows you to locate it on a map and remotely delete all your data. When you get it back, you can restore everything from your last backup.

C. Series 40:

Series 40 is Nokia's mid-tier embedded software platform and designed in Java, it's like a customized version. This was introduced in 1999, third generation is introduced in 2005 and its fast growth clicked in 2011-12 period. The key features of this operating system are simplicity, responsiveness and speed. Its major disadvantages are not supporting multitasking and do not have native code application interfaces for third parties. So this OS does not support the applications not written in Java.

D. Symbian:

Symbian is an open source (from 2008) operating system mainly used in Nokia Mobile Devices. The underlying OS was historically created by Symbian Ltd and licensed by Nokia and other phone manufacturers. Symbian OS is designed to make minimal demands on batteries and to have low memory. It is a multitasking operating system and very less dependence on peripherals. All applications are designed to work seamlessly in parallel. The use of technologies based on agreed- upon standards is a basic principle of Symbian OS, ensuring that applications are robust, portable, and interoperable. Memory management optimized for embedded software environment. Application support for international environment with built-in Unicode character sets. Symbian uses microkernel approach. The kernel manages system resources such as memory and is responsible for time-slicing the applications and system tasks.

Safety and security: The integrity and security of user data is of paramount importance. Symbian OS offers gate keeper type security. The system asks user permission to install any applications. There are three concepts, which are the foundation of Symbian OS platform security architecture. Tiers of Trust: A mobile phone tends to be used by one person only; this is particularly true of smart phones which hold personal information such as contact details and calendar entries. The design of Symbian OS assumes this Trusted Computing Base: It controls the lowest level of the security mechanisms and has the responsibility for maintaining the integrity of the system. The trusted computing base includes the operating system kernel, which looks after the details of each process, including the set of privileges assigned to it. Some Symbian OS phones are 'closed', that is they do not support installation of native add-on software; on such a closed phone, the kernel, including the kernel-side device drivers and the file server are the only fully-trusted components.

Trusted Computing Environment: This consists of further trusted software provided in the mobile phone by Symbian. TCE code usually implements a system server process – failure of one server should not threaten the integrity of the operating system itself: the

kernel can restart the server and maintain that integrity. Each server has limited privileges to carry out a definite set of services. By not granting all privileges to all servers, Symbian OS limits the threat exposed by any flaw or corruption of a server. By requiring servers to have certain privileges, it is possible to limit access to sensitive low-level operations to selected servers and, thereby, prevent misuse of these operations by other processes.

E. Windows :

Windows is the most popular computer operating system. Past five years they are started to give more attention on mobile operating system also. It is offering new user interface with 'Metro' design. They designed Windows CE (Compact Edition) specifically for handheld devices, based on Windows API. Later introduced Windows 7 version. Recent version Windows 8 mobile OS released at June 2012, its support, many of great features like multi core processor support, hi-fi screen resolution, higher storage support and near field communications. This mobile OS is almost simulating the personal computer version of Windows 8. Windows 8.1 mobile version is progress and it's targeting the mobile markets in 2014.

Safety and Security: Windows operating systems are mainly taking care Device encryption, Data Encryption, Data Leak Prevention and Digital Signature.

- Device Encryption: Full internal storage encryption to protect information. It is built on Windows Bit Locker architecture.
- Data Encryption: It helps provide privacy and authentication between two communicating parties who have exchanged a shared secret.
- Data Leak Prevention (DLP): Information Rights Management (IRM) helps prevent intellectual property from being leaked. It helps to protects emails and documents on the phone from unauthorized distribution. Easy to deploy on Exchange Server and SharePoint Active Directory Rights Management supports all your Mobile Information Management (MIM) needs. Digital Signature: It helps to authenticate another party, or information sent by that party, without prior exchange of a shared secret.

F. Blackberry :

The BlackBerry OS is the proprietary mobile platform developed by RIM (Research in Motion), exclusively for its BlackBerry smart phones and mobile devices. It offers native support for corporate mail via MIDP, which enables effortless wireless sync with Microsoft Exchange, Lotus Domino and email, contacts, calendar, notes and so on, while used along with the BlackBerry Enterprise Server. This OS additionally supports WAP 1.2. Its network architecture is differing than other operating systems.

Blackberry provides end to end encryption. It is using two encryption options. Advanced Encryption Standard (AES) and Triple Data Encryption Standard (Triple DES). Data sent to the BlackBerry smart phone is encrypted by BlackBerry Enterprise Server using the private key retrieved from the user's mailbox. The encrypted information travels securely across the network to the smart phone where it is decrypted with the key stored there. It enabled RSA SecurID Two-Factor Authentication. Additional authorization also available when users access application data or corporate intranets

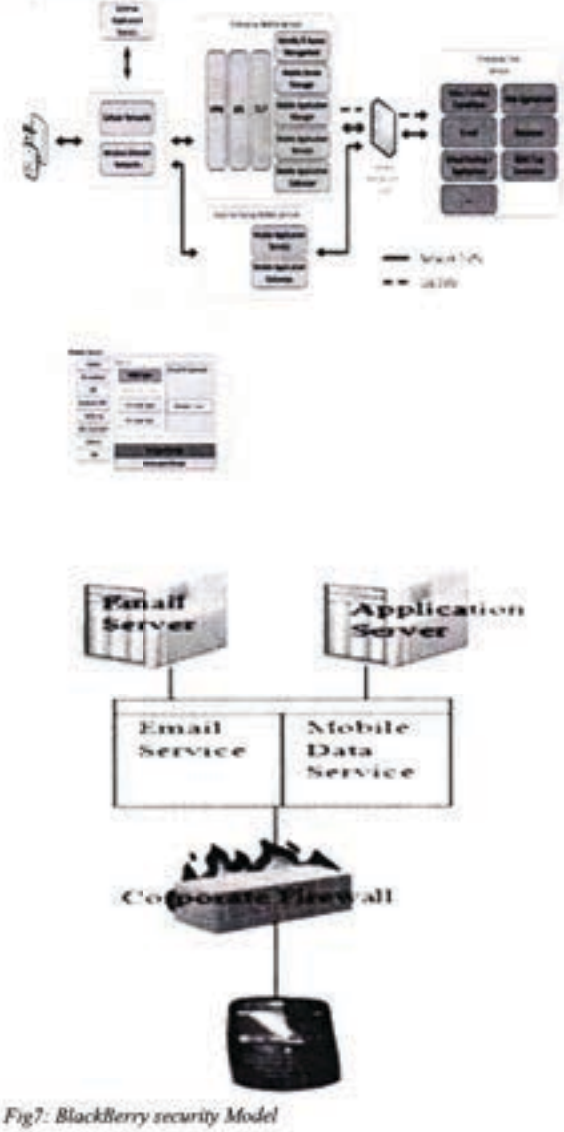


Fig7: BlackBerry security Model

III. SECURITY MEASURES

Importance of mobile security measures are increasing day by day. Now many of personal and professional information is stored in mobiles. Mobile Os are under target to attackers as other operating systems. Recent days mobiles users are highly demanding the internet features. Many of the users are wish to carry single compact device for multiple activities.

As per MSRA (Mobile Security Reference Architecture) provides an architecture pattern that can use to ensure the confidentiality, integrity, and availability of data accessed through a mobile computing solution.

Major mobile security measures require in mobile operating system.

a) Authorization
b) Device Management
c) Identity and Access management.
d) Data management
e) Logging
f) Personnel and Facilities Management
g) Network Access Control
h) Software Validation
i) Patch Management

When choosing a mobile based on operating system, consumers can validate the required security functions are available on their device. Device management includes OS configuration, software patches, data management etc. Based user identity access activities have to enable. The Logging security is includes various policies and infrastructure management. When installing new software it has to

Validate whether it is provided by genuine software provider. Required periodic patches update to maintain the device as secure.

As per mobile security experts the analysis reveals today's mobile platform have wide difference. Device management, application security and corporate e-mail support are almost fine but still required improvements. Below bar charts are giving a brief idea on various security measures between Android, iOS, Windows and blackberry comparison on built-in security, data protection, authentication, device protection, application security, mobile device management, corporate managed e-mail, device firewall, security certifications, virtualization.

IV. RESULTS AND FINDINGS

Based on the analysis and data collected, we can finalize best option. The analysis is giving a brief idea about the market trends on mobile operating systems. The present trend leads Android and iOS operating systems in market.

- Android is a freeware operating system from internet giant Google. Its base version itself provides many of the features, which gives lot of popularity. Many of mobile companies did their own research and customization on top of this and making that more popular. As it as wide users and development peoples huge number of software are available in market. This also provides a better memory management, process management, and network options. As Android OS is an open platform many of them doing OS level development, it is security challenge.
- Apple derives the iOS operating system, which is popular in mobile software perfection and professionalism. Multiple mobile thoughts they contributed to mobile technology. Many of other operating systems are following their features. Comparatively iOS is giving better security than android.
- Few of other operating systems are gives greater security. Blackberry maintains separate security architecture. Many of them in using blackberry OS for professional activities.

V. CONCLUSIONS

So, which platform should you buy from a security standpoint? For most users the answer will be iOS, but for the technically experienced Android can work if they are careful. However, if a user is willing to jailbreak they can get many of Android's benefits anyway. Blackberry may be a good choice from a security standpoint, but generally those who want a consumer device will prefer the others for non-security reasons. Windows Phone and the other platforms may be good in future, but at present there probably has not been enough exposure to make this risk a good long term bet.

Platform	Platform Development Rate and Deployment	Flexibility	Security Usability	Platform Security	Ecosystem Security	Privacy and Data Leakage concerns	Ranking
iOS	OK	Poor	Great	OK	OK	OK	1st
iOS (Jailbroken)	OK	High	OK	Poor	Moderate	Poor	2nd Equal
Android	High	High (Very Good)	OK	Poor	Moderate / Poor	Poor	2nd Equal
Blackberry	Low / Moderate	Low (Inflexible by design)	OK	High	High	Great	2nd

In short, recommendation for each type of phone user:

- Non-technical person: iOS (iPhone/iPad/iPod touch)
- Techie: iOS/Android
- Business user: Blackberry / iOS (but check what the company standard is first)

VI. REFERENCES

[1] 'Statistical reports reference for various operating systems' http://stats.areppim.com/
[2] 'Statistical reports reference for mobile and personal computer http://www.smartinsights.com/
[3] 'Msec Research Group' https://www.msec.be/
[4] 'Mobile Security Reference Architecture' https://cio.gov/
[5] 'Enterprise Readiness of Consumer Mobile Platforms' http://www.trendmicro.com/

Selection of Programming Language

Guided By: Sudarshan Sirsat

Sandeep Kamath, Aashil Karkoria, Ankur Kanoujiya, Angha Baikar, Afzal khan, Mukesh Koli, Jay Kotecha, Kurtika Hariharan, Akshay Mathur, Shashikant Maurya

Abstract— **Selection also called a decision, one of the three basic logic structures in computer programming. The other two logic structures are sequence and loop. In a selection structure, a question is asked, and depending on the answer, the program takes one of two courses of action, after which the program moves on to the next event. This structure is sometimes referred to as an if-then-else because it directs the program to perform in this way: If Condition A is True then perform Action X else perform Action Y. All logic problems in programming can be solved by forming algorithms using only the three logic structures, and they can be combined in an infinite number of ways. The more complex the computing need, the more complex the combination of structures.**

I. INTRODUCTION

"A GOOD PROGRAMMING LANGUAGE IS A CONCEPTUAL universe for thinking ABOUT PROGRAMMING"ALAN PERLIS

NATO CONFERENCE ON SOFTWARE ENGINEERING TECHNIQUES ROME, 1969. Programming language research covers a broad spectrum from systems work through to theory. On the theoretical side programming language research hasits roots in the work of Schon_nkel, Curry, Kleene, and Church in the 1920s and 1930s. On the practical side, it has been inuenced by the developments in hardware spanning from the von Neuman architecture of the 1950s to the heterogeneous distributed networks of today. Also developments in software engineering with its requirements of feasibility, reliability and performance have had great impact on the area. A feature of programming language research that gives it much of its vibrancy is that individual researchers have the opportunity to work across broad bands of this spectrum. The area is mainly concerned with the discovery of principles and generic techniques; it is not, with a few notable exceptions, about the invention of new languages. Many of the results of the area are generally applicable to a wide variety of language paradigms including procedural languages, functional and logic languages, object-oriented languages etc. Many common programming language features have emerged from this area; some of the more well-known

II. DIFFERENT PROGRAMMING LANGUAGES

A. C

C is one of the most widely used programming languages and often used as an introduction to programming. It has influenced many languages that came after it, and knowledge of C will make learning later languages, such as Objective-C (used by Apple), easier. It influences many later languages you could want to learn, so starting with C will give you a deeper understanding of how computers work.

B. Java

Java is a higher level language which is designed to be compatible with any operating system. It has similar syntax to C and C++. It's a great programming language to start with because it is widely used and practical, however it won't give you as deep of an understanding of computer operation as a lower level language like C will.

C. C++

C++ bridges the gap between a language like C and Java as it has features of both low-level and high-level languages. It's another commonly used language that has a wide range of uses and compatibility. It's based off of C and adds object-oriented features. It has also influenced many other languages such as C# and Java.

D. Python

Python is a language that was designed with human readability in mind. Because of this, it doesn't take as much code to execute programs as other languages. It's a great, easy way to learn recurring concepts in computer science and has real world use in the creation of scripts.

E. Ruby

Ruby has similar function to Python but is less readable. It's more object-oriented than Python and is similarly designed with simplicity in mind. It has many applications, but is most often used for web applications.

F. HTML and CSS

HTML and CSS are used for webpage design. While these languages won't really help pave the way for learning more traditional programming languages, they are essential for webpage design. HTML (Hypertext Markup Language) is a "markup language" which allows you to put content into a webpage whereas CSS (Cascading Style Sheets), is used to format and define the layout of a page.

G. MIT App Inventor for Android

If you aren't interested in programming as a profession (at least at the moment) it may be worth looking at using the MIT App Inventor for Android. It requires no coding, but will teach you how programmers think and provide knowledge on some concepts in computing. Plus, you'll end up being able to make Android apps once you've mastered it!

H. What's Next ?

If you already have knowledge of another programming language then these are great follow-up languages.

I. C#

C# is primarily used for Windows applications in the .NET Framework. Learning C# is easy if you have experience in C, C++, or Java. The syntax is similar. Its popularity has been increasing as C# is used for third-party apps on Windows 8 or Windows Phone.

J. Objective-C

Objective-C is primarily used for Apple's operating systems, OSX (for Macs) and iOS (for iPhone and iPad). If you are looking to develop for Mac, Objective-C is the way to go. Apple provides lots of support for learning Objective-C through their developer program.

K. Javascript

JavaScript (little relation to Java) is a common language used to make webpages more dynamic. With syntax similar to C, it doesn't require a lot of effort to set up as it's built into web browsers. It's also used in other applications such as PDFs.

L. PHP

PHP is another language often used for web development, although it works well as a general-purpose language as well. PHP can be implemented directly into HTML. Those looking to learn PHP should already know HTML, CSS, and JavaScript.

III. BASICS OF SELECTION OF PROGRAMMING LANGUAGE

A. Technical Characteristics

There are plenty of technical characteristics with which to compare programming languages: how many key words, maximum length of identifier, type-checking facilities, polymorphism, overriding, and so on. It is almost infinitely perplexing. Instead, I suggest that you focus on the following comparison criteria, which are centered on outcomes:

- Ease of learning
- Ease of understanding
- Speed of development
- Help with enforcement of correct code
- Performance of compiled code
- Supported platform environments
- Portability
- Fit-for-purpose.

B. Ease of Learning

The Clearly, the easier that the programming language is to learn, the quicker that programmers become productive. Java is a lot easier to learn than C++; C is probably even easier. But you learn once and program for a long time, so that ease of learning is of only limited value.

About the most difficult language to learn that I know is C++. One of the weird things about C++ is that the language is so complex and flexible that you develop your own style and work mostly within a subset of the language. This is strikingly illustrated by the two Microsoft object libraries: MFC (Microsoft Foundation Classes) and ATL (Active Template Library), which have a large overlap in functionality, but look completely different. Learning to be proficient in C++ is best seen as a three-step process: basic understanding, developing your own style, and learning to read someone else's style. Classes and books teach you only step one; but it is only when you have passed step two that you are useful. You will be of more help to others when you have completed step three. It is probably the same for all languages; but the time that is taken in steps two and three is much shorter in languages that are not as profuse in features as C++.

C. Ease of Understanding

Most code is written once and read many times—usually, to focus on a particular point (for instance, to fix a bug). Thus, it is important that the reader quickly grasp the essence of what's happening. COBOL can usually be read easily; but, because it is verbose, you have to read many lines of code to get anywhere. Old-fashioned COBOL (in contrast to OO COBOL) tended to use PERFORMs, instead of procedure calls, and that means that the logic and the data are miles apart. C does not have this problem; but C can still be hard to understand. This is partly the fault of the language. For instance, I always have to think twice to remember that:

<code>int* ar [20]

Is an array of pointers and not a pointer to an array of integers?

In theory, object-oriented (OO) languages allow you to write more compact code (because of code reuse), and the structure of the objects can allow you to mirror more closely the structure of the problem; thus, in theory, they should be easier to understand. In practice, you might find many abstract classes; and you have to spend time trying to figure out what they are all for, before you can identify all of the parts of the program that are relevant to the topic that you are investigating.

Programmers can make any program hard to understand. Unfortunately, there is a breed of programmers who think that cryptic code is good code—especially, C and C++ programmers. It used to be the case that cryptic code produced fast, compact applications; but compiler optimizers are so good these days that this is no longer an excuse.

D. Speed Of Development

If you look at speed of development in the round, you must consider not only how long it takes you to write code, but also how long it takes you to find a solution to the problem at hand

and find the bugs. Factors other than the programming language—for instance, platform facilities, development tools, experience and skill of the programmers, and testing regime—are so significant that it is hard to pin down any difference in development speed that is actually due to use of different programming languages. For instance, a quick calculation will show you that the physical act of typing code makes practically no difference to the speed of development. Verbose, therefore, does not mean unproductive.

There is a programming language called APL (initials for A Programming Language) that takes compactness to the extreme by using a host of additional graphic symbols, and was best used if you had a specialized keyboard. It was described to me once as a "write-only programming language"; having been written, the code is immediately incomprehensible. Not surprisingly, the language has fallen out of fashion.

Object orientation has probably made good programmers better and bad programmers worse. Usually, with OO programming, there are more ways of tackling a problem than in conventional languages—or, at least, there seems to be. This means more time up front thinking about the design. Once that is done, however, development speed should be faster—mainly, because of the reuse opportunities.

The dream of reusable class libraries from which classes can be extracted and glued together by non-experts never materialized. Creating the classes was no problem. Finding and reusing the classes was the hard part, because you had to find a class that was both a solution to the problem and did not have unwanted side effects, such as using other classes, databases, or system facilities that you did not have or did not want. One of the messages from the agile community is: Only program for reuse, when a reuse opportunity arises. In other words, do not develop classes that you think will be suitable for reuse. Instead, refactor the code to create a reusable class only when you want to reuse it. I find this to be good advice.

One of the simplest forms of reuse is to copy code from one place and paste it in another. The benefits of OO reuse kick in when you want to modify the code; now, you modify it only in one place. In a poorly designed OO program, you find that it is hard to modify the code for one subclass without having to break the functionality of another subclass. In a well-designed OO program, each piece of shared code has a clear task, and making it better benefits all subclasses. Finding the right solution can be hard and is often an iterative process, which is why refactoring is so important.

E. *Help with Enforcement of Correct Code*

The ideal programming language should turn logic errors into syntax errors. A powerful means to this end is type checking. Most standard languages, such as Java and COBOL, have good type checking. But some languages have an escape clause. An example is C++, in which you can change the type of a pointer. Figure 1 shows an example.

```
char a[1000];
class C {int x; int y;};
C* p = (C*)a + 50;
p->x = 100;
```

Figure 1. Changing the type of a pointer

This example creates a pointer to an object of type **C**, but points it 50 characters into array **a**. Clearly, this feature completely circumvents type checking, and it is considered dangerous. But I have had hardly any trouble by using this feature—possibly, because I know it's dangerous and, so, tread carefully.

Most languages have pitfalls of their own. These are caused by being able to write two expressions that look similar on paper, but have very different effects. For instance, in C and C++: <code>if (A = B) X(); means that A is assigned the value of B, and X is called if A is not equal to zero; while: <code>if (A == B) X();means to call X if A equals B. Java is safer, because the first expression gives a syntax error.

F. *Performance Of Compiled Code*

These days, performance is as much an architectural issue as a programming issue. Thinking of performance problems about which I have heard over the last five years, there was one that used a workflow tool for all of its application logic. It used far too many I/Os. There was another that did not upgrade its network when it moved from old-fashioned terminals to a Web interface. And there was another that had a program that would cache hits to searches, but, every now and then, clear its cache and grind to a halt rebuilding it.

I cannot remember a recent case in which bad performance came down to inefficient code production by the compiler. Even with a games program that is processor-bound, the chances are that a large percentage of the processing is going toward drawing the screen. That, in turn, means that performance comes down as much to how you use the graphics card as how the compiler generates code. Of course, huge, number-crunching programs will be an exception; but, even there, only a small part of a large program is performance-critical.

G. *Supported Platform Environments*

By platform environment, I mean not only the operating-system facilities, but also the middleware facilities, database facilities, and system-management facilities. Clearly, the more facilities that you have, the more work that has already been done for you. However, there is a downside. Understanding the platform facilities to the level that you can use them wisely is more difficult, in my opinion, and takes longer than understanding the programming language. Furthermore, I find that I use a facility, make it work, and move on. I do not revisit the subject for months or years. The effect of this is that I am much more dependent on documentation, examples, and

research by using the Web than I am with normal programming.

H. Probablity

Most popular languages have been standardized by a no vendor organization. The aim is twofold: to reduce retraining needs and enhance portability. However, portability has been found to be very difficult. The standards bodies have succeeded in making differences of syntax a relatively minor problem, but that only covers the part of the language that is standardized. Most languages are dependent on hardware constraints in some form, such as defining the maximum value that an integer can take. If code is used such as seen in Figure 1, the dependency between code and platform can be tight and hard to unravel.

Probably, however, the most serious problem with portability is the platform environment. For instance, the problems of moving a mainframe COBOL program to a .NET environment will almost certainly be in the area of changing from—say, a CICS interface to a .NET interface. By far, the most successful example of a popular language that has good portability is Java, which was deliberately designed for portability. It has achieved this by standardizing not only the language, but also the platform environment (J2EE and J2SE).

I. Fit-for-Purpose

While Java is a good language and is highly portable, it is unsuitable for some purposes, such as some game programming and system programming. Game programming often requires fast access to the screen-display hardware. This can be done by using DirectX, which is available only in C++ or C. The kind of code that Figure 1 illustrates might not be portable, but it is highly useful if you must take a buffer of information and unpick the data from it. You need this if you are writing your own database or network software. However, the majority of programmers are writing business applications in which neither of these issues applies, and Java (or C# or Microsoft Visual Basic) is just fine.

Historically, there have been many good programming languages that got nowhere in the market, simply because they did not have good interfaces to the platform environment. This was probably true of Pascal, for instance, and it helps explain why that never became a popular commercial language.

J. Decisions

So, how do you decide which programming language to use? The first overriding factor is the last one in the preceding list: fit-for-purpose. For most business operational systems, you need a language that has good middleware and database facilities. For specialized work—for instance, writing a hardware driver—you might be forced to use C or C++. For some AI work, perhaps, a language such as Lisp or Prolog might be the only one that gives you access to the facilities that you need.

The second factor is platform choice. If your application must integrate with other applications, you will often find it easier to implement if it is written in the same language as those other applications. For instance, calling a Java method by using Java RMI is easier from another Java program. Calling a .NET application method is easier from another .NET application. Also, if your program must create a batch file for an existing COBOL program, it will be easier and quicker to write the new application in COBOL—especially, if you can use existing COBOL copy libraries.

The third major factor is the existing skills base of your programmers. Training an existing Java programmer to write in C# would probably take a few days, because the languages are similar. What would take much longer would be training your programmers in the .NET environment. For the first few months of their new life as C# programmers, they will be looking things up in the reference manual, asking each other how such-and-such worked, and spending time pondering (and, we hope, properly investigating) the best way to perform a new task. As time progresses, much of this additional effort disappears, and both productivity and quality creep up.

The net result of the last two factors is that choosing a programming language becomes a strategic skills issue for the IT department. You want to concentrate your resources on a few languages. You want to grow skills in languages that you think will have long-term benefits for your organization. However, you do not want to do it so fast that you are left with all of the programmers struggling with a new language.

IV. CONCLUSION

The three major factors that influence the choice of programming language are the following:

- The language must be fit-for-purpose. Normal business-application development can be done in Java or COBOL, but specialist applications might need C++ or some other language.
- The choice of platform is critical. It's better to let the platform dictate the programming language, instead of letting the programming language dictate the platform.
- The skills of the programmers. Programmers are much more productive when they are working in a language that they know well, instead of working in a new language or on a new platform.
- Is there—or will there be—a shortfall in programming expertise?

V. REFERENCES

[1] guidance on choosing a programming languagechris britton-http://msdn.microsoft.com/en-us/library/cc168615.aspx

[2] Best programming languages to learn on your own time. http://www.techrepublic.com/blog/career-management/best-programming-languages-to-learn-on-your-own-time/

Research Article on Software Sizing Techniuqes

Guided by: Sudarshan Sirsat

Narayan Mehta, Anuradha Mishra, Ruban Nadar,Johny Johnson, Nishit Mohanan, Indal pal, Neeraj Mourya, Spandan Jain

***Abstract*— Software sizing is an activity in software engineering that is used to estimate the size of a software application or component in order to be able to implement other software project management activities (such as estimating or tracking). Size is an inherent characteristic of a piece of software just like weight is an inherent characteristic of a tangible material. Historically, the most common software sizing methodology has been counting the lines of code written in the application source.**

Keywords— estimate, tracking; inherent; weigh; lines of code; software application;

I. INTRODUCTION

Historically, the most common software sizing methodology has been counting the lines of code written in the application source. Another sizing method is the IFPUG method called Function point analysis. The IFPUG FPA functional sizing method (FSM) has been used successfully, despite being less accurate estimating complex algorithms and is relatively more difficult to use than estimating lines of code. Variations of Function Points include MK II Function Point, NESMA Function Points, Object Oriented Function Points, OOFP, and newer variants as Weighted Micro Function Points which factor algorithmic and control flow complexity. For more information about the similarities and differences between these ISO FSM methods see IFPUG and COSMIC - Similarities and Differences. The best Functional Sizing Method depends on a number of factors, including the functional domain of the applications, the process maturity of the developing organization and the extent of use of the FSM Method There are many uses and benefits of function points[3] beyond measuring project productivity and estimating planned projects, these include monitoring project progress and evaluating the requirements coverage of COTS (Commercial off the shelf) packages.

Other software sizing methods include Use Case based software sizing, which relies on counting the number and characteristics of use cases found in a piece of software and COSMIC which addresses sizing software that has a very limited amount of stored data such as 'process control' and 'real time' systems.

Both the IFPUG Method and the COSMIC Method are ISO/IEC standards.

IFPUG method to size the non-functional aspects of a software or component is called SNAP. The non-functional size is measured by SNAP Points. The SNAP model consists of four categories and fourteen sub-categories to measure the non-functional requirements. Non-functional requirement are mapped to the relevant sub-categories. Each sub-category is sized, and the size of a requirement is the sum of the sizes of its sub-categories. The SNAP sizing process is very similar to the function point sizing process. Within the application boundary, non-functional requirements are associated with relevant categories and their sub-categories. Using a standardized set of basic criteria, each of the sub-categories is then sized according to its type and complexity; the size of such a requirement is the sum of the sizes of its sub-categories. These sizes are then totaled to give the measure of non-functional size of the software application.

II. SOFTWARE SIZING TECHNIQUES

There are many models for software estimation available and prevalent in the industry. Researchers have been working on formal estimation techniques since 1960. Early work in estimation which was typically based on regression analysis or mathematical models of other domains, work during 1970s and 1980s derived models from historical data of various software projects. Among many estimation models expert estimation, COCOMO, Function Point and derivatives of function point like Use Case Point, Object Points are most commonly used. While Lines of Code (LOC) is most commonly used size measure for 3GL programming and estimation of procedural languages, IFPUG FPA originally invented by Allen Alrecht at IBM has been adopted by most in the industry as alternative to LOC for sizing development and enhancement of business applications. FPA provides measure of functionality based on end user view of application software functionality. Some of the commonly used estimation techniques are as follows:

- *Lines of Code (LOC):* A formal method to measure size by counting number of lines of Code, Source Lines of Code (SLOC) has two variants- Physical SLOC and Logical SLOC. While two measures can vary significantly care must be taken to compare results from two different projects and clear guideline must be laid out for the organization.
- *IFPUG FPA:* Formal method to measure size of business applications. Introduces complexity factor for size defined as function of input, output, query, external input file and internal logical file.
- *Mark II FPA:* Proposed and developed by Mark Simons and useful for measuring size for functionality

in real time systems where transactions have embedded data

- *COSMIC Full Function Point (FFP):* Proposed in 1999, compliant to ISO 14143. Applicable for estimating business applications that have data rich processing where complexity is determined by capability to handle large chunks of data and real time applications where functionality is expressed in terms of logics and algorithms.

- *Quick Function Point (QFP):* Derived out of FPA and uses expert judgment. Mostly useful for arriving at a ballpark estimate for budgetary and marketing purposes or where go -no go decision is required during project selection process. Object Points: Best suited for estimating customizations. Based on count of raw objects, complexity of each object and weighted points.

- *COCOMO 2.0:* Based on COCOMO 81 which was developed by Barry Boehme. Model is based on the motivation of software reuse, application generators, economies or diseconomies of scale and process maturity and helps estimate effort for sizes calculated in terms of SLOC, FPA, Mark IIFP or any other method.

- *Predictive Object Points:* Tuned towards estimation of the object oriented software projects. Calculated based on weighted methods per class, count of top level classes, average number of children, and depth of inheritance.

- *Estimation by Analogy:* Cost of project is computed by comparing the project to a similar project in the same domain. The estimate is accurate if similar project data is available.

III. ESTIMATING SOFTWARE SIZE

An accurate estimate of software size is an essential element in the calculation of estimated project costs and schedules. The fact that these estimates are required very early on in the project (often while a contract bid is being prepared) makes size estimation a formidable task. Initial size estimates are typically based on the known system requirements. You must hunt for every known detail of the proposed system, and use these details to develop and validate the software size estimates. In general, you present size estimates as lines of code (KSLOC or SLOC) or as function points. There are constants that you can apply to convert function points to lines of code for specific languages, but not vice versa. If possible, choose and adhere to one unit of measurement, since conversion simply introduces a new margin of error into the final estimate. Regardless of the unit chosen, you should store the estimates in the metrics database. You will use these estimates to determine progress and to estimate future projects. As the project progresses, revise them so that cost and schedule estimates remain accurate.

The following section describes techniques for estimating software size.

Developer Opinion
Developer opinion is otherwise known as guessing. If you are an experienced developer, you can likely make good estimates due to familiarity with the type of software being developed.

Previous Project Experience
Looking at previous project experience serves as a more educated guess. By using the data stored in the metrics database for similar projects, you can more accurately predict the size of the new project .If possible, the system is broken into components, and estimated independently.

Count Function Blocks
The technique of counting function blocks relies on the fact that most software systems decompose into roughly the same number of "levels". Using the information obtained about the proposed system, follow these steps:

1. Decompose the system until the major functional components have been identified (call this a function block, or software component).

2. Multiply the number of function blocks by the expected size of a function block to get a size estimate.

3. Decompose each function block into sub functions.

4. Multiply the number of sub functions by the expected size of a sub function to get a second size estimate.

5. Compare the two size estimates for consistency.
Compute the expected size of a function block and/or a subfunction with data from previous projects that use similar technologies and are of similar scope.

Function Point Analysis
Function points allow the measurement of software size in standard units, independent of the underlying language in which the software is developed. Instead of counting the lines of code that make up a system, count the number of externals (inputs, outputs, inquiries, and interfaces) that make up the system.

There are five types of externals to count:
1. External inputs - data or control inputs (input files, tables, forms, screens, messages, etc.) to the system.

2. External outputs - data or control outputs from the system

3. External inquiries - I/O queries which require a response (prompts, interrupts, calls, etc.)

4. External interfaces - libraries or programs which are passed into and out of the system (I/O routines, sorting procedures, math libraries, run-time libraries, etc.)

5. Internal data files - groupings of data stored internally in the system (entities, internal control files, directories)

Apply these steps to calculate the size of a project:
1. Count or estimate all the occurrences of each type of external.

2. Assign each occurrence a complexity weight.

3. Multiply each occurrence by its complexity weight, and total the results to obtain a function count.
Complexity description of Low, Medium and High are:
External inputs 3 4 6
External outputs 4 5 7
External inquiries 3 4 6
External interfaces 5 7 10
Internal files 7 10 15

4. Multiply the function count by a value adjustment multiplier (VAM) to obtain the function point count. Where Vi is a rating of 0 to 5 for each of the following fourteen factors .The rating reflects how each factor affects the software size.

1. Data communications
2. Distributed functions
3. Performance
4. Heavily used operational configuration
5. Transaction rate
6. On-line data entry
7. Design for end user efficiency
8. On-line update of logical internal files
9. Complex processing
10. Reusability of system code
11. Installation ease
12. Operational ease
13. Multiple sites
14. Ease of change

Assign the rating of 0 to 5 according to these values:

0 - factor not present or has no influence
1 – Casual influence
2 - Moderate influence
3 - Average influence
4 - Significant influence
5 - Strong influence

Function point analysis is extremely useful for the transaction processing systems that make up the majority of MIS projects. However, it does not provide an accurate estimate when dealing with command and control software, switching software, systems software or embedded systems.

Count Externals

Counting externals is the application of function point analysis (presented in the previous section) to real-time embedded systems. Instead of a function point count, the end result is an estimated size in KSLOC. This is based on estimated counts of the following externals:

- external inputs
- external outputs
- external inquiries
- □ external interfaces

The procedure for estimating the size of a new project in KSLOC is as follows:
1. Identify the number of program externals for each major program component (function block or software component). Either estimate all four externals, or work with these three externals: inputs, outputs, and inquiries.

2. Apply one of the following formulas:
KSLOC = 12.288 + 0.030E (where E is the sum of all four externals)
KSLOC = 13.94 + 0.034A (where A is the sum of the three externals)

If the KSLOC and externals information is stored in a metrics database from previous
Projects, you can develop specific formulas to more accurately reflect an organization's software development potential. Do this by plotting size (y-axis) versus counts of externals (x-axis), and fitting a line that best represents the data. The formula that describes this line is the estimating formula.

Combining Estimates
A good method for improving the accuracy of estimates is to estimate several ways, and then calculate a weighted average of the estimates. For example, to average four estimates obtained by different techniques:
1. Assign each estimate a weight (such that all the weights sum to 1.0). A small weight indicates confidence in the estimate while a large weight indicates uncertainty in the estimate.
2. Compute the weighted size for each estimate (size *weight).
3. Sum the weighted sizes to get the combined estimate.
Estimates for a result of 129 KSLOC.

IV. RECOMENDATION OF ESTIMATING SIZE

Estimate the software size using a number of techniques, and then average these results to produce a combined estimate. As the metrics program matures, use the data collected from previous projects to develop specific estimating procedures and formulas. Remember to re-estimate as the project progresses. Software estimates usually increase over the life of the project, and you should adjust cost and effort estimates accordingly.
Estimate the size of each program component (function block or software component) independently and relate this size to similar products and project components.

V. CONCLUSION

The accurate prediction of software development costs is a critical issue to make the good management decisions and accurately determining how much effort and time a project required for project managers as well as system analysts and developers. There are many software cost estimation methods available including algorithmic methods, estimating by analogy, expert judgment method, top-down method, and bottom-up method. No one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other. To understand their strengths and weaknesses is very important when you want to estimate your projects. For a specific project to be estimated, which estimation methods should be used depend on the environment of the project. According to the weaknesses and strengths of the methods, you can choose some methods to be used.

VI. REFERENCES

[1] *Guidance on How to Choose a Functional Size Method - Pam Morris Total Metrics - Function Point Resource Centre see Iso/Iec 14143-6: - Software Engineering — Software Measurement — Functional Size Measurement — Part 6: Guide For Use Of Iso/Iec 14143 Series And Related International Standards*

[2] *Uses and Benefits of Function Point Counts - Pam Morris Total Metrics - Function Point Resource Centre.*

[3] *Uses and Benefits of Function Point Counts - Pam Morris Total Metrics - Function Point Resource Centre*

Cyber Crime and Underground Economy

Guided By: Dr. Vinita Gaikwad

Swati Verma, Peter Sequeira, Yogija Prabhu, Arunima Bhol, Lorna Lopes

***Abstract*—Cybercrime represents an underground economy of $114 billion. It's organized, employs expert hackers and operates like any legitimate economy. MBAs and other business experts can choose big targets for cybercrimes, just as they might plan the strategies a business employs. On the other side of the equation, the cybersecurity industry is in a rut, practically guaranteeing the continuing operation of cybercriminals. Let's take a look at how the underground Internet economy operates.**

Keywords—underground internet technology, cybercrime, industry, marketing, information.

I. Introduction

Technology is invading our lives. Every day we perform a multitude of operations using computing devices; we check out bank account balance, we pay parking and we exchange documents with our colleagues and friends. Mobile, social networks and cloud computing are the paradigms that have changed the online user experience; these platforms manage today almost all of the information in the internet, an impressive and priceless amount of data.

The information is money and criminals follow it, the schemes of monetization are various and target every sector. Cybercrime is evolving in complexity and organizational capacity, under many aspects it works exactly as a major enterprise.

Cyber-attacks, malware, identity theft, phishing and spam are the emerging threats that menace the users in cyberspace. The attackers commit malicious activity that creates a shadow economy comparable to the one of principal states of the planet.

As usual, the most interesting information on cybercrime activities are provided by the principal security firms that collect data from global intelligence networks able to detect and analyze cyber threats identifying emerging trends in malicious activities. Symantec Security firm publishes annually a very interesting report titled the "Internet Security Threat Report", a complete overview on the principal cyber threats detected by its network during the year. In 2011 Symantec detected and blocked over 5.5 billion malware attacks, an impressive figure with an increase of 81% compared to the previous year, and in many cases the offensives remain undetected for long periods. The situation is worrisome, lack of awareness on principal cyber threat, poor information security in design, users risky behaviors and economic crisis are all factors that concurs to exacerbate the situation.

Cybercrime is assuming an amazing relevance in our lives. What really surprises me are its organizational capability that has nothing to envy to the most efficient industry. It is assuming a typical hierarchical structure in which every actor has a well-defined role and responsibility. From leaders to money mules, cybercrime is practically infiltrating every sector of society. The hierarchy's "executives" oversee operations, define the strategy and the business model to implement, verifying that everything proceed as planned. Core of criminal business is the technology, groups of specialists that are able to deploy sophisticated malware, arrange private botnets, and design fake artivirus software as well as efficient exploit toolkits.

Just like any legitimate organization, the code is reviewed and subject to a strict validation processes. Another interesting aspect is the recruiting process for large-scale operations in which specialized affiliates set up recruitment programs searching for specific technologic profiles to arrange cyber-attacks.

Cyber criminals use to the following to promote their products/services: Internet job boards, hacking message forums, and underground IRC chat channels.

You might expect that the costs of getting into the cybercrime business are high, but it's not nearly as expensive as you might think. Most of the tools and data that a cybercriminal needs can be picked up on the cheap. Hiring a botnet can be a luxury item at $225, but a keystroke logger is about $20 and website hosting for a phishing scam can be as low as $10. Even labor costs can be low: many of these crimes no longer require an expert hacker. Someone with just a little technical knowledge can go shopping online for the tools to commit cybercrimes on an almost automatic basis. Commonly, a criminal will purchase a set of tools to steal information and then turn around and sell that information to someone else to exploit, just as legal businesses might form supply chains.

The cybercrime economy exists somewhat openly: there are web-based forums and other websites where cybercriminals list tools and information for sale and discuss future projects. An outsider could almost think he was visiting any legitimate industry forum. While marketing both the tools of the cybercrime trade and the information gathered through different exploits is not as simple as promoting a legal enterprise, sites are full of offers.A few years ago, most efforts to commit crimes online were directed toward breaking into Windows-based computer systems. Now, hackers are targeting smartphones and tablet systems that are commonly run with minimal protection against viruses and malware, at least compared to desktop computers. This targeting is especially concerning because mobile platforms are becoming one of the key areas where many people and organizations handle financial transactions.

The most popular type of information for sale through these underground markets is credit card details, since it is much easier to make a purchase online using a credit card than it is to drain a bank account or steal an entire identity. That being said, bank account information is also a popular commodity. The cybercrime economy is thriving during a time when legitimate enterprises are struggling. The costs of dealing with

these crimes are mostly paid by large organizations, such as banks, and both prevention and damage control require larger investments in cybersecurity. But even with the new initiatives, laws and tools for combating cybercrime, those costs are going to continue to grow. There are numerous organizations and individuals suggesting that cybercrime may get significantly worse in the years to come, especially in light of how easily the techniques and tools of cybercriminals can be turned to cyberwarfare and cyberterrorism. The only solution is for the cybersecurity industry to catch up with its illegal counterpart.

II. ECONOMIC IMPACT OF CYBERCRIME

The number of attacks blocked by Symantec is just the tip of the iceberg and cybercrime is surely responsible for the majority of them. The privileged weapons used are the malware, malicious code of increasing complexity that in many cases have eluded security mechanisms, principal vectors of attacks exploited by cybercriminals are web attack kits and socially engineered phishing techniques.

Targeted attacks exploiting zero-day vulnerabilities are the most dangerous, they aren't the most common. The use of exploits of known vulnerabilities for which a slow, and often incorrect patch management process exposes IT infrastructures to concrete risk to be compromised.

III. DATA COLLECTED

The InfoSec Institute CISSP Training course trains and prepares you to pass the premier security certification, the CISSP. Professionals that hold the CISSP have demonstrated that they have deep knowledge of all 10 Common Body of Knowledge Domains, and have the necessary skills to provide leadership in the creation and operational duties of enterprise wide information security programs.

InfoSec Institute's proprietary CISSP certification courseware materials are always up to date and synchronized with the latest ISC2 exam objectives. Our industry leading course curriculum combined with our award-winning CISSP training provided by expert instructors delivers the platform you need in order to pass the CISSP exam with flying colors. You will leave the InfoSec Institute CISSP Boot Camp with the knowledge and domain expertise to successfully pass the CISSP exam the first time you take it. Some benefits of the CISSP Boot Camp are:

A. Dual Certification - CISSP and ISSEP/ISSMP/ISSAP

We have cultivated a strong reputation for getting at the secrets of the CISSP certification exam

Our materials are always updated with the latest information on the exam objectives: This is NOT a Common Body of Knowledge review-it is intense, successful preparation for CISSP certification.

We focus on preparing you for the CISSP certification exam through drill sessions, review of the entire Common Body of Knowledge, and practical question and answer scenarios, all following a high-energy seminar approach.

The following is some data provided by Symantec in the report related to malicious activity related to cybercrime:

Web based attacks increased by 36%, with over 4,500 new attacks each day. 403 million new variants of malware were created in 2011, a 41% increase of 2010. SPAM volumes dropped by 13% in 2011 over rates in 2010. 39% of malware attacks via email used a link to a web page. Mobile vulnerabilities continued to rise, with 315 discovered in 2011.

Looking at the data, it's easy to foresee a further increase of the phenomena also for 2012.

In October 2012, Ponemon Institute Research, Sponsored by HP Enterprise Security, published the report "2012 Cost of Cyber Crime Study: United States" that quantifies the economic impact of cybercrime on the US economy, and the report proposes worrying scenarios. Cybercrime impact has been estimated in reaching $8.9 million in 2012, and average annual cost increased 6 percent in respect to the previous year, an impressive growth that has been driven up by attacks on websites, denial-of-service attacks and malicious insiders.

Companies are daily victims of different type of attacks such as data breaches, system destruction and violation of networks. It has been estimated that there are an average of 1.8 successful attacks each week.

The US has suffered the largest cost cost, followed by Germany with $5.9 and Japan at $5.1. Cybercrimes are very costly for organizations, despite the organizations size. All industries fall victim to a different degree.

IV. RESEARCH & OBSERVATIONS

A. Cybercrime: A secret underground economy

Cybercriminals are making a killing off of stolen identities, creating their own market for buying and selling credit card and bank account information on the cheap.

EMAIL | PRINT | SHARE | RSS

By David Goldman, CNNMoney.com staff writer

LAST UPDATED: SEPTEMBER 17, 2009: 9:11 AM ET

Cybercriminals sell your information on Internet Relay Chats such as this one. One line reads 'Selling US/Ca& worldwide Cc's for the best prices."

Cybercriminals can see what you enter on your screen and steal your credit card information or bank account information.

B. How To Be Safe Online:

- Think ID theft can't happen to you?
- Staying ahead of the hackers
- How to be a (safe) Wi-Fi warrior

.NEW YORK (CNNMoney.com) -- If the word 'cybercrime' conjures up images of computer geeks trying to crash computers from their mothers' basements, think again.

Cybercrime has become a rapidly growing underground business built by savvy criminals, who buy and sell valuable stolen financial information from millions of unsuspecting Internet users every year in an on online black market.

"Most cybercriminals are very, very interested in financial gain by compromising customer accounts," said FBI special agent Austin Berglas, who supervises the Bureau's New York Internet crimes squad. "Believe it or not, there are people who fall victim to their scams, and we see it every day."

Because cybercriminals are so skilled at hacking into thousands of computers every day, the crime is potentially a billion-dollar business. If every stolen credit card and bank account had been wiped clean last year, that would have netted cybercriminals some $8 billion, according to data from Symantec, maker of the Norton antivirus software.

As a result of the lucrative payout, more and more online criminals are entering the game. In fact, the number of new Internet security threats rose nearly three-fold last year to 1.7 million.

Those cyber attacks mostly come from malware, or malicious software, that hands control of your computer, and anything on it or entered into it, over to the bad guys without you even knowing it. The most common forms of malware include keystroke logging, spyware, viruses, worms and Trojan horses.

How the deed is done? Once your information has been stolen, cybercriminals go onto an invitation-only Internet Relay Chat (like a chat group) to do commerce with other online criminals. Cybercriminals will often set up a hacker channel for a matter of days, do business, and then take it down to avoid detection. When active, hacker IRCs can get upwards of 90,000 cybercriminals talking to one another at a given time, according to Dave Cole, senior director of product management at Symantec.

Online criminals use the IRCs to sell or trade your credit card or bank account information. Credit cards are some of the cheapest commodities sold on the Internet Black Market, averaging about 98 cents each when sold in bulk. A full identity goes for just $10.

Credit cards and bank account information made up 51% of the goods advertised on the underground economy last year, up from 38% in 2007. Credit cards are most popular because they're the cheapest stolen commodity. Cards with expiration dates, CVV2 numbers and names go for more than ones with numbers only, but there is no honor in the underground online crime world -- oftentimes hackers will sell the same credit card information to multiple users, and many have already been canceled.

As a result, buyers and sellers on IRC channels will often give the information to a trusted third party for a fee. The third party will test the card information, often by charging a very nominal amount or by posing as a charity, and then verify the goods to the buyer.

After the information is purchased by a secondary criminal, that person can use a machine to print out a fake credit card with your information. But many use yet another tertiary person to wire stolen money into an overseas bank account.

That third person in the chain is usually called a "mule," who often doesn't even know he or she is part of an underground organized crime scheme. Many mules respond to the "make money from home" schemes, where stolen money is sent to their accounts, and they subsequently wire that money to an overseas account for a 10% to 15% fee.

Other mules are given phony ATM cards and are asked to retrieve cash for a small fee. But there is substantial risk involved -- law enforcement usually comes knocking on mules' doors first.

To catch a thief. The FBI is working undercover in many of these IRC channels in an effort to thwart the cybercriminals. And in many cases, captured criminals agree to work for the government in exchange for reduced sentences.

"After we make an arrest for someone cashing out at ATM machines, I'll tell them they can go to jail for 10 years or they can come work for Team America," said Berglas.

The strategy doesn't always work. Albert Gonzalez, the infamous TJ Maxx (TJX, Fortune 500) thief who stole 45 million credit card numbers and private information of 450,000 customers in 2007, was an FBI informant. He helped bring down a massive credit card theft scheme, but double-crossed the FBI, using insider information to help fellow criminals evade detection and carry out the TJ Maxx theft.

Security software also helps, but it far from solves the problem. To avoid detection, many cybercriminals will send out just a handful of viruses before modifying the code and sending it out again.

"The truth is that 'fingerprint' security technology is no longer effective," said Rowan Trollope, senior vice president of product development at Symantec. "The bad guys that got involved are organized professionals, and they figured out how to get around our technology."

Though Trollope said the new version of Norton's antivirus software helps address the problem byscanning for files' reputations, he said that Internet consumers also need know how how to keep their identities safe online.

"We do products really well, but the next step is education," said Trollope. "We can't keep the Internet safe with antivirus software alone."

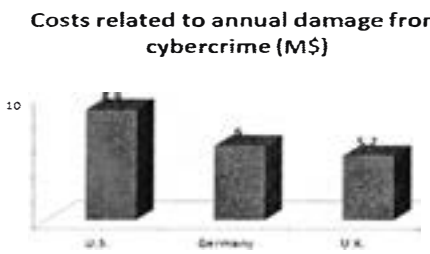


Figure 1 – Total cost of cybercrime 2012 – Ponemon Institute

Small organizations pay a higher per capita cost than respected big enterprises. Ponemon Institute estimated in 2012 a cost per seat of $1,324 for small businesses and of $305 for enterprises. The industries with the highest cost for cybercrime are defense, utilities and energy and financial service.

The cyber-attacks which cause major economic losses are denial of service (DOS), malicious insider and web-based attacks. They represent the 58 percent of all cybercrime annual costs per organization, the research demonstrated that the impact of cybercrime is higher if mitigation solutions are applied late.

C. *Cybercrime, A Social Threat*

Governments, intelligence agencies and law enforcement agree on the consideration that cybercrime represents one of the primary threats to the security of collectivity; it has equal and devastating impact on industries and private citizens. The "2012 NORTON CYBERCRIME REPORT" provided interesting elements to evaluate the incidence of cybercrime in the daily users experience online.

The scale of consumer cybercrime is amazing – 1.5 million daily victims that amounts to an annual global cost of $110 billion. Four out of ten social network users have fallen victim to cybercrime but the data that is more worrisome is that one out of six users report that someone has committed a cyber-attack to their profile, typically it is hacked by someone that pretend to impersonate them. Social network platforms and mobile platforms are considered privileged channels for scams and frauds; a new wave of malware is targeting the powerful platforms and their users.

The highest numbers of cybercrime victims are located in Russia, China and South Africa, countries where malicious activities are very popular.

The data demonstrate the efficiency of cybercrime activities and justify the concerns of security community and governments, criminals are evolving their techniques adapting them to the evolution of IT tools.

Figure 2 – Geographic location of victims (Symantec)

D. *Deep In The Cybercrime Underground*

It's impossible to provide a clear picture of the wide worldwide hacker community, a complex universe composed by a wide number of groups that share opinions, tools and any other information on dedicated and self-managed forums. The platforms are various; every technology is good if the final intent is sharing, even better if it is possible to keep away law enforcement and other prying eyes.

These places were are also used for illegal activities such as the sale of malicious code and the providing of hacking services, a black market that is reaching impressive figures.

Many security companies have tried to infiltrate these groups, they represent an incredible source of information and their monitoring could allow to discover the cybercrime trends and to detect new cyber threats and new dangerous customization of well know malware.

V. CONCLUSION

The fight to cybercrime is an arduous task an endless clash between the law enforcement and cyber-criminal groups that are growing under organization aspects and that are able to provide products and services more and more advanced. Cybercrime is a worldwide phenomenon that menaces economics and the security of every state. Governments and law enforcements must cooperate to face with cyber threats.

Sharing data on criminals' events is a crucial step. Information must be proposed to common people through an awareness campaign. Private business and governments must collaborate to monitor criminal activities detecting them as soon as possible via the monitoring of internet, especially of underground forums.

The detection of criminal activities could be improved using security intelligence systems such as SIEMs, the definition and the share of security best practices is another aspect that could limit the level of penetration for cybercrime in private business and moderates the cost of cybercrime, let's think for example to the possibility to avoid spear phishing attacks due a proper awareness campaign.

Of course, all these reasons must be supported by the establishment of a legal framework accepted globally that severely punishes cyber criminals everywhere they operate. Unfortunately, we are still far from these conditions that are absolutely necessary to cope with an industry that knows no crisis and that is showing frightening growth.

VI. REFERENCES

[1] *http://www.group-ib.com/index.php/rassledovanie/30-link-investigate*

[2] *http://securityaffairs.co/wordpress/3612/cyber-crime/reflections-on-the-zero-days-exploits-market-starting-from-forbess-article.html*

[3] *http://now-static.norton.com/now/en/pu/images/Promotions/2012/cybercrimeReport/2012_Norton_Cybercrime_Report_Master_FINAL_050912.pdf*

[4] *http://www.ponemon.org/local/upload/file/2012_US_Cost_of_Cyber_Crime_Study_FINAL6%20.pdf*

[5] *http://securityaffairs.co/wordpress/3913/cyber-crime/1-day-exploitsbinary-diffing patch-management-the-side-threats.html*

[6] *https://www.europol.europa.eu/sites/default/files/7th_enisa_cert_workshop_report.pdf.pdf*

[7] *http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-101.pdf*

Malware Security And Threats

Guided by: Dr. Vinita Gaikwad
Sushant More, Chintan Kinderkhedia, Rahul Darji, Shameeka Paradkar, Sana Shaikh

Abstract: **Lately, a new kind of war takes place between the security community and malicious software developers, the security specialists use all possible techniques, methods and strategies to stop and remove the threats while the malware developers utilize new types of malwares that bypass implemented security features. In this study we closely looked into malware, to understand the definition, types, propagation of malware, and detecting/defending mechanisms in order to contribute to the process of protection and security enhancement.**

Keywords: ***Malware, Propagation, Detection, Honeypot, Obfuscation, Privacy***

I. INTRODUCTION

With the escalating growth of communication and information systems, a new term and acronym invaded the digital world called as malware. It is a general term, which stands for malicious software and has many shapes (codes, scripts, active content and others). It has been designed to achieve some targets such as, collecting sensitive data, accessing private computer systems, even sometimes harming the systems. The malware can reach the systems in different ways and through multiple media; the most common way is the downloading process from the internet, once the malware finds its way to the systems, based on the functions of the malware the drama will begin. In some cases, the malware will not totally harm the system, instead affect the performance and creates overload process; in case of spying, the malware hides itself in the system, which cannot be detected by the anti-virus software, these hidden malware send critical information about the computer to the source. Based on the above challenges, it is critical to carry out an in-depth analysis to understand the malware for better detection and removal chance. This paper is organized as follows: section two has covered the recent state of the malware security and threats through results obtained from different journals. Section three discusses about the types of malware, section four presents the malware analysis techniques. Section five studies the propagation of malware in different applications and environment, and finally section six explains malware detection techniques.

II. RECENT STATE OF MALWARE SECURITY AND THREATS

Technology has become an element key for today's life style where both business and research worlds completely rely on the technology and its applications. However like the other side of the coin, these developments have also opened the doors for the hacking and attacking community, and within a few years the malware has become a major security threat, affecting computers and networks widely.Initially, the hackers and attackers started invading others computers just for fun they did not have any serious intention to look for any great gains, until online commerce gained its popularity especially in banking, financial transactions etc, which made the hacker to get financial gains.This has motivated the attackers, to work more and more to keep the machines infected as longer as possible, to get more financial gains and more valued information and data. Consequently a big challenge has emerged in terms of protecting the information and business systems and a kind of arm races have started between security products and attackers community. The malware historical timeline shows that it has a lot of changes and phases since it has been discovered and detected in hosts and

networks , starting from virus which is a self-replicating malware but not self-transporting , moving to worm, which is a self-replicating and self- transporting, and going more for other malware types and families . With the rapidly increasing complexity and interconnection of emerging information systems, the number of malware attacks is also increasing piercingly.

While, there are a noticeable development in defense technologies and security techniques, there is also a similar development in sophisticated hacking techniques and appearance of new security vulnerabilities from day to day. Due to the sequence of malware propagation, we can now clearly feel the impact of malware on various computer network infrastructures, technologies and services such as, file-sharing, online social networking Bluetooth and wireless Networks .Many techniques have been developed and used to detect malware and prevent its propagation like sandboxing and virtual environment and some time the malware environment has been simulated to make it easy to detect by using FRAM model .The enhancement and improvement process for security should be powerful and simultaneously move in two directions; protecting the systems from the well-known malware threats and seeking for innovative ideas and insightful analysis for handling the malware issues.

1. *Malware Issues*

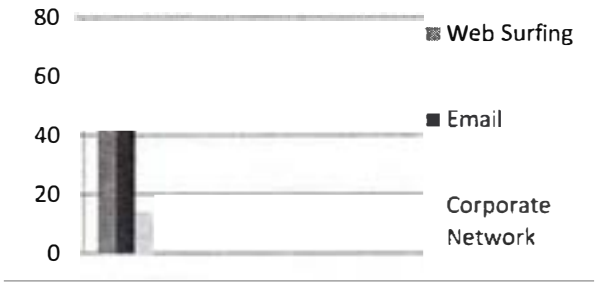
Many studies, surveys, experiments, brainstorming, statistical analysis and modeling methods have been done to gain deeper knowledge and valuable information

about malware , because the attackers are continually developing their abilities, attacking skills and techniques. In order to make the tracking and detection processes difficult, and to pose new challenges to inspectors, all these studies and works are not sufficient enough to cover the rapid increase in malware evolution. Based on our understanding Virus Bulletin (1988) was the first dedicated Journal to study the malware, while, now there are a lot of Journals available that are dedicated to the security issues, especially malware issues.

2. *Data Collection*

New findings from an Osterman Research survey show that organizations are showing little, if any, improvement in deterring malware from entering their organization. And it's attacking them from all sides - via the web, over email, and through increasingly embraced social media and Web 2.0 applications.

The survey, which polled security-focused decision makers and influencers and was sponsored by Trustwave, found:



- 74 percent of organizations have been infiltrated by malware through web surfing over the past year.
- Another 64 percent have experienced malware infection through email.
- And 14 percent have had malware enter their corporate network through social media or other Web 2.0 applications.

Meanwhile, most are finding that their ability to deal with these risks, especially web threats, is staying the same or getting worse. Traditional solutions don't seem to be helping.

III. MALWARE SECURITY

Recently, the number of information security threats caused by malware has rapidly increased, which leads to urgently studying the threats and accordingly categorizing them, to simplify the process of discovering and handling them, in order to detect them and find appropriate solutions. Malware has been categorized into seventeen different types , in this section we have listed and discussed the main and most common categories as follows:

Virus: is a computer program that has the ability to harm and self-replicating in order to infect host; viruses are linked or attached to a software utility (e.g. PDF document). Launching the infected PDF document could then activate the virus, and a sequence of events may occur based on the function of the virus.

Worm: another kind of harmful programs is worm, which can replicate itself and invisibly transfer through networking. The effects of worms differ from viruses as the former need help from any file, to work and mainly its effect is on networking bandwidth or sending junk emails. One example of worms is Conficker.

Spyware: this may occur, when users download free or trial software. In this kind, the users are observed by spies; hence their passwords, account numbers and every other personal detail become vulnerable.

Adware: this kind usually happens, while downloading free games or it is combined and embedded with advertisements, so when we watch advertisements this embedded code is installed to our PCs. This kind aims to observe the user's activities, when using networking.

Trojan: this kind gives power to remote hijackers, to use your system as they wish. They may get your passwords, observe your systems or damage the system files.

Botnet: this kind of malware controls your systems remotely and sends spam or spyware. Most of botnets are zombie and wait for command of the party who runs it, where there are two types of botnet such as, simple or hierarchical.

IV. MALWARE PROPAGATION

Many studies and researches focused on studying the malware propagation in the digital world, communications and computer networks, some of the modeling and experimental procedures have been followed to study the effect of malware and the way it propagates in these fields, in addition to this, the studies cover some concepts and techniques related to malware detection. The malware propagation concept refers to the electronic method, by which, malware is transmitted to an information system, platform or device it seeks to infect for example the malware can propagate through PDF files and access the host unless the user disable the JavaScript in PDF reader

1. Through Operating System

Malware is attacking the operating systems such as Mac, Android, Windows and Linux, but not in the same level and strength because some operating systems have more defense mechanisms which don't allow the malware to achieve its design purpose. In the following lines some attacks followed by malware against OSs will be highlighted to show how the operating systems act accordingly. Every year a large number of new OSs

malware with stronger propagation and strategies are created.

The malware follows dynamic and adaptive propagation to attack the OS architecture such as, attack the security levels in OSs to open security threat. Another malware method propagates the OSs to infect the executable file and create virtual tasks which will slow down the OS performance. Propagation of malware differs, based on OS, for example, the malware work on (.plist) Macintosh system files, but in Android it comes as spyware which attack the source code of Android OS .

2. Through Wireless Networks

Bluetooth technology has been introduced in specific project named as Blue Bag that includes a covert attack and scanning device, which demonstrates how attackers can infect and reach a wide range of mobiles and devices running a Bluetooth Technology, they have found some weaknesses in Bluetooth technology, which may allow attackers to reach the devices. In the authors have explained some specific attacks that can affect the wireless communication and Bluetooth such as:

BlueSnarf: it uses the (Object exchange) push service and the attacker can access without any authentication and recently in the upgraded version of this kind of attack, the attacker can get a full access including read and writes access.

Bluejacking: occurs by sending a short tricky text message into authenticated dialog, and the users will be using the access codes of the tricky message, which allows the attacker to take control of the device.

BlueBug: the attacker will be able to use phone services, which include incoming and outgoing calls, sending and receiving SMS, etc. all through accessing the cell phone.

Blue Bump: it goes through the weakness of Bluetooth in the way it handles link keys, and it can lead to getting the data or abusing the mobile services such as internet, WAP and GPRS.

Blue Smack: it simply guides to service denial.

HeloMoto: it is a combination of BlueBug and BlueSnarf effect.

Blue Dump: the attacker will involve himself in the pairing process through Bluetooth after dumping the stored link key.

Car Whisperer: the default configuration of some devices makes the PIN code fixed for pairing and exchanging, which will make it easy for the attackers, to abuse the devices and take control of the devices accordingly once they get the PIN, which is not changeable.

Blue Chop: the attacker will get the chance to is connect and terminate the established connection, especially when the master of the connection is supporting multiple connections. Rresults after conducting the survey as follows:

1) Bluetooth technology is involved in many devices cell and Smartphones, PCs Notebooks, GPS Printers, palm pilots and others, which means more possibility for malware to propagate.

2) Visibility time is an important factor in the possibility of being attacked, longer time more possibility, and unfortunately some users are not aware about this point and hence keep Bluetooth on discoverable and visible mode in need and without need.

3) Social Engineering factor: 7.5% of the owners are simply careless in terms of the received files and they tend to accept the unknown files from unknown sources.

4) The survey shows that, small percentage of people are aware about the risks that they may face, when they use the new technology devices, such as, smart phones, and how this can affect their work and organizations, where the data value is high and critical, if they save it on their devices, and it is possible that, they can carry the risk with them to their organizations and work.

V. MALWARE DETECTION TECHNIQUES

Since malware has different types, behaviors and different level of risk, the same detection methods and mechanisms cannot be used in all cases. It is impractical to have just one security software to efficiently handle the malwares. Hence having different detection methods for different environments becomes unavoidable. This study had focused on the most common and powerful techniques such as honeypot, honeynet, virtualization (partial and full), sandboxing and behavior operation sets. A massive experiment had been done by Taiwan malware analysis net (TWMAN), it was based on virtualization concept and client-server model, the experiment added a great value to the field of malware detection since it was able to detect many malwares which were not detectable by normal detection methods, going forward, we can clearly see that the detection process needs more computer processing power and advance techniques to make sure that the nature and behavior of malware are clear and covered from all the angles and views.

1. Anomaly-Based

Anomaly-based detection looks for unexpected or abnormal behavior indicators, which indicate the presence of malware. In more detail, anomaly based detection creates a baseline of expected operation. After this baseline has been created, any different form of baseline is recognized as malware. We have identified that the anomaly based detection technique uses the previous knowledge of what is known as normal to find out what is malicious. A special type of anomaly based detection techniques is specification based detection. A specification based detection uses set of rules to determine what is considered as normal, with the purpose of making a decision about the maliciousness of the program that breaches the rule set. The basic limitation of the specification based system technique is the difficulty to correctly determine the program or system behavior .

2. Honeypots

Honeypots are security devices whose value lie in being probed and compromised. Traditional honeypots are servers (or devices that expose server services) that wait passively to be attacked. *Client Honeypots* are active security devices in search of malicious servers that attack clients. The client honeypot poses as a client and interacts with the server to examine whether an attack has occurred. Often the focus of client honeypots is on web browsers, but any client that interacts with servers can be part of a client honeypot (for example ftp, ssh, email, etc.).

There are several terms that are used to describe client honeypots. Besides client honeypot, which is the generic classification, honeyclient is the other term that is generally used and accepted. However, there is a subtlety here, as "honeyclient" is actually a homograph that could also refer to the first open source client honeypot implementation (see below), although this should be clear from the context.

Architecture

A client honeypot is composed of three components. The first component, a queuer, is responsible for creating a list of servers for the client to visit. This list can be created, for example, through crawling. The second component is the client itself, which is able to make a requests to servers identified by the queuer. After the interaction with the server has taken place, the third component, an analysis engine, is responsible for determining whether an attack has taken place on the client honeypot.

In addition to these components, client honeypots are usually equipped with some sort of containment strategy to prevent successful attacks from spreading beyond the client honeypot. This is usually achieved through the use of firewalls and virtual machine sandboxes.

Analogous to traditional server honeypots, client honeypots are mainly classified by their interaction level: high or low; which denotes the level of functional interaction the server can utilize on the client honeypot. In addition to this there are also newly hybrid approaches which denotes the usage of both high and low interaction detection techniques.

3. Malware Behavior

Behavior based detection techniques study and analyzes the behavior of suspected or known malicious code, such as destination and source addresses of this code, and the way in which, the code was attached. Behavior based detection technique differs from the other scanning techniques as it considers the action performed by the malware, rather than the binary pattern. The programs with different binary content but having same behavior are collected. These types of detection techniques help in detecting the malware, which keeps on generating new signature versions, because they will always use the recourses of the system in the same manner. The behavior detector collects the data, interprets the data, and then applies the matching algorithm .

About 67% of malware produces sub-process when executed. Some malicious behaviors appear after malware execution, like thread injection and self-delete. These malicious behaviors are called as Malicious Behavior Feature (MBF). The term Behavior Operation Set (BOS), which defined by file actions (e.g. read, rename), process actions (e.g. terminate, create), network action (e.g. TCP, UDP), and registry actions (e.g. open key, query value). These four operations were used to extract and investigate the behavior.

FRAM model had been proposed to make a malware forensic repository for the purpose of malware analysis. FRAM is mixed from open source tools and commercial tools which integrated together to propose an automated system. This automated system aims to reduce the time needed to handle the new malware and increase the rate of success reverse engineering malware. In MalTRAK the users can run any program without asking for any policy or rules, but MalTRAK guarantee that the user can recover the clean state if the infectious state were found. MalTRAK can satisfy this by storing many logical views during the program run time. The draw back with the MalTRAK model that the extra overhead in disk space and run time, but using this model we have a very good recovery result to clean state in case of infection..

4. Malware Signature

Normal antivirus software look for signatures, which are a sequence of bytes in the malware code to state that if the program scanned, is malicious or not. Essentially, there are three types of malware: basic, polymorphic, and metamorphic malware. In basic malware, the malware developer changes the entry point of the program. Polymorphic viruses alter themselves, while leaving the original code unchanged. A polymorphic virus contains an encrypted malicious code beside the decryption part. This virus is enabled by a polymorphic engine, which is included in the body of the virus. The polymorphic engine generates new versions every time it is run; thus it is very difficult to detect this type of virus by signature based detection techniques. Metamorphic malware use advanced obfuscation techniques, to reprogram itself therefore the children and parent signatures are very different. It is not possible to detect this type of malware without disassemble the virus file.

There are many problems associated with the signature based detection technique. The biggest problem is that, the signature generation is a very complex process and requires a strong code analysis algorithm. The second problem is that the signatures are distributed as fast as possible. The third problem is that, new signatures can easily bypass the detectors, and the final problem is that, the size of signatures repository is increasing day by day.

5. Obfuscation and Normalization

It is a technique used by software developers and writers targeting to hide the details of their products so that the reverse engineers can't find the correct code, it has been used as an advantage by the malware writers to achieve the same goal, obfuscation can be achieved by different operations and easily can make changes in the signature of malware in order to make the process of detecting the malware very difficult.The obfuscation techniques can be done in different methods, starting from inserting some (no operation) instructions and inserting (push-pop) x, which known as dead-code because nothing will be achieved and accomplished and inserting some instructions for branching unconditionally, moving to inserting process for some registers and substituting instructions, all of these methods will guide to obfuscate the code of the malware and make the process of detection difficult to malware scanners. Malware normalization can be identified as a process and mechanism to detect the obfuscated copies of malware and increasing the rate of catching the malware by the detector, the output of the normalization will be the original signature of the malware which has been obfuscated and accordingly the signature will be compared to the signatures to verify it, then it will be saved in the list of known signatures in order to decrease the time of scanning and detecting next times.

The normalization process can be done through some steps as follows: decompressing the binary code of malware, then disassembled it and pass it to the normalizer to eliminate the obfuscation and get the original code, finally passing the normalized code to malware detector in order to get out the signature, compare it with the available list and get the matched one.

Since the signatures of malware are long and take many comparisons times to detect them there was a need for additional procedures such as API Procedure to reduce the time of normalization and detection process. The key to enhance the process of malware detection based on signatures is via developing better disassembler and better algorithm for analyzing the similarity.

VI. CONCLUSION AND FUTURE WORK

The malware developer tries to write new techniques and strategies to hide the malicious code and infect the targets. On the other hand, the detectors analyze malware behaviors continuously and try to resist these techniques and strategies hence, we need to allow detection development techniques to lead malware updating through very well analytical process for malware activities and behaviours to fix any possible targeted threats. A new simulation must be designed to contain real system samples, to analyze the malware behaviours against these samples after elaborate malware updating. The objectives of this simulation are to avoid systems threats before being infected by real malware.

VII. REFERENCES

[1] malwaremustdie.org/

[2] sdiwc.net/digital-library/web-admin/upload-pdf/00000875.pdf

[3] https://www.europol.europa.eu/sites/default/files/7th_enisa_cert_workshop_report.pdf.pdf

[4] en.wikipedia.org/wiki/Malware_research

[5] http://www.enigmasoftware.com/malware-research/

[6] http://en.wikipedia.org/wiki/Category:Types_of malware

[7] http://www.pandasecurity.com/homeusers/security-info/types-malware/

ABOUT THE INSTITUTE

Thakur Institute of Management Studies, Career Development and Research was established in the year 2001 with a clear objective of providing quality technical education in tune with international standards and contemporary global requirements, offering 3 years postgraduate degree in Master of Computer Applications (MCA). The Institute is recognized by the AICTE norms and is affiliated to the University of Mumbai.

The Management's commitment to excellence is reflected in the marvelous infrastructure that is comparable to the finest institution of its type in the country. The sprawling campus with lawns, gardens, playgrounds, parking area, hostel accommodation and temple ensures a right academic ambience essential for a center of higher education.

At TIMSCDR, the importance of faculty is well understood which is reflected in qualified and experienced teaching staff. A closely monitored quality, assurance mechanism ensures proper coverage of syllabus within right time frame.

Application of modern technology in teaching-learning process and day to day governance of the Institute makes TIMSCDR unique. The organization supported by dedicated 16 Mbps broadband internet connectivity and also has WI-FI facility.

The Institute focuses on imparting knowledge to the students that persists even when they pass out and step into the corporate world. The syllabus has been given a new dimension through experienced faculty and state of the art infrastructure. The overall personality development through extra curricular activities like quiz, debates and seminars to name a few have been a hallmark of the Institute.

Thakur Educational Trust's (Regd.)

THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER DEVELOPMENT & RESEARCH

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village, Kandivali (E), Mumbai - 400 101

• **Tel:** 6730 8301, 02, 28840484/91 • **Telefax:** 28852527

Email : timscdr@thakureducation.org • **Website :** www.timscdrmumbai.in • www.thakureducation.org