

I.E.S. College of Engineering

2nd Internal Examination

Date : 23 April 2020

Name : Jovial Joe Jayarson

Roll No. : IES17CS016

Subject : CS-308 · Software Engineering & Project Management

Marks Awarded:

A1.) Code Review

- ~ Code review is a systematic examination, which can find and remove vulnerabilities in the code.
- ~ These vulnerabilities include - memory leaks, buffer overflows, security breach etc.
- ~ Software review is an essential part of software development life cycle (SDLC) that helps to validate software engineers the quality of functionality and other vital features of the software.
- ~ This usually performed manually but nowadays advanced AI powered tools are available to enhance and speedup the process.

Code Review Techniques

① Checklist

- ~ The most common and probably the best reading technique is by using a simple check list.
- ~ This list contains questions that help developers focus upon certain areas of the code.

② Perspective based Reading

- ~ It is another technique that enables the ~~pro~~ developer to analyze the program/software in the user's perspective.
- ~ This will help them to see the part of the picture that was previously hidden.

③ Using Advanced Tools

- ~ Code Editors like VSCode and Atom, provide integrated (or as plugins) code review features.
- ~ These are powered by software specifically designed to analyze the code and suggest changes (eg: - in linting, optimizing, unused variables etc).
- ~ These tools further boost the process of code review.

A2.)

Function Point

- ~ A function point is a component of a software development which helps to approximate the cost of development in the process.
- ~ It is a process which defines the required functions and their complexities in a piece of software in order to estimate the software's size and scope upon completion.

How calculation is done

- ~ A function point calculate software size of the software with the help of logical design.
- ~ It also includes performance of functions as per user requirements.

Step1: There is a scale of complexity adjustment factor

$$F = 14 * \text{scale}$$

scale — 0 \Rightarrow No influence

1 \Rightarrow Incidental

2 \Rightarrow Moderate

3 \Rightarrow Average

4 \Rightarrow Significant

5 \Rightarrow Essential

Step 2: Calculate the Complexity Adjusted Factor (CAF)

$$CAF = 0.65 + (0.01 * F)$$

Step 3: Calculate the Unadjusted Function Point (UFP)

From the Table

<u>Function Units</u>	<u>Low</u>	<u>Average</u>	<u>High</u>
External Input	3	4	6
External Output	4	5	7
External Query	3	4	6
ILF Inserts } Data functionality	7	10	15
ELF	5	7	10

~ Multiply each individual function point to the corresponding values.

Step 4: Calculate the Function Points

$$FP = UFP * CAF$$

A3) ~ Phases of Validation Testing includes α -Tests & β -Tests

α -Tests

~ Carried out by the customers at the developers side under supervision of the developer

β -Tests

~ Carried out by the end user at the customer side and does not involve developer.

~ Carried out after system testing

~ Carried out alongside system testing

~ Performed by testers in pairs of the customer representative

~ Performed by ~~test~~ test but tester may or may not be present

~ Performed in production environment.

~ Performed in development environment.

~ Data used may be defined by the testers with the help of a test case.

~ Data is generally ~~used~~ guessed - not constrained.

A4.) Cyclomatic Complexity

~ It is a known complexity measurement method or technique.

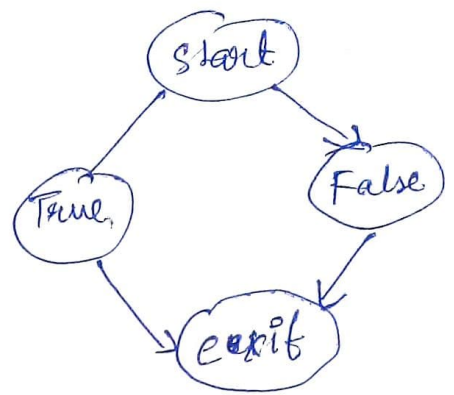
~ It measures the number of linearly independent paths through a program's source code.

~ If a code has a single if else statement then, there would be two paths:

- (i) where the if condition is true
- (ii) where the if condition is false.

~ It is depicted with the help of a control flow graph.

eg:- if (condition) {
 \equiv
 }



~ Now the graph has 4 nodes and 4 edges.

~ The cyclomatic complexity ~~with~~ of a graph with N - nodes & E - edges ~~are~~ is $E - N + 2$

$$\therefore \text{here cyclomatic complexity} = 4 - 4 + 2 = 2$$

Range

Less complex $\Rightarrow 1 < \text{cyclomatic complexity} < 2$

Complexity $\Rightarrow 5 < \text{cyclomatic complexity} < 7$

If cyclomatic complexity > 10 it is invalid and the code must be refactored.

A5.) Ripple Effect

~ A ripple effect is a situation in which an effect from an initial state can be followed outwards in a chain.

~ This is just like the ripples that are caused from the epicenter of the stone-drop in the water.

- ~ Recent software major feature mods have predicted impact analysis and accounts for ripple effect as one of their stages.
- ~ One of the best example is the ~~trampa~~ incompatibility of python 3 and 2.
- ~ When the developers of python 3 decide to introduce certain new features of ~~py~~ it became backward incompatible.
- ~ This then rippled across the future versions of Python 3 and ~~there~~ therefore it had to branch off completely from Python 2 development.
- ~ Another example is that whenever we try to fix a security hole in Java script based applications it is very likely that the effect is seen in the other part of the software.
- ~ Therefore it is a must to test a software after each and every minor change that can potentially break the software.

A 6.) Adaptive maintenance

- ~ This type of maintenance as a result of external influences.
- ~ This is also affected by strategic change brought within the company.
- ~ The system ~~try~~ tries to adapt to the new changes to survive.

* For example:

- ~ The customer suddenly ~~for~~ require a new mechanism to perform payment.
- ~ This shift in requirement forces the developer into a power-play mode.
- ~ It means that whatever requirement there is, it must be satisfied - quickly but at the same time without breaking what should be smoothly functioning.
- Therefore the developer designed the new change from ground up to the satisfaction of the customer.

- Another example would be where certain restrictions are imposed by Government.

- ~ To survive the developer will have to block the requested (by govt.) services until further notification.

(eg:- Amazon - pauses delivery on non-essential items)

A7.) Four P's of Software Project Management

- ~ For effective software management the four following P's are crucial.

(i) People (ii) Product (iii) Process (iv) Project

(i) People

- ~ This is what constitutes a software project management it included various categories of people.

- ~ They are, Managers, Stakeholders, Developers, Customers etc.

- ~ They help in retain the product its market value and status.

- ~ They function from different levels and places but with a single purpose.

(ii) The product

- ~ Before a full fledged project is planned a product is usually visualized.
- ~ These include its objective scope, technical details & constraints all are identified.
- ~ Without this information it is impossible to determine a reasonable cost, amount of risk, project schedule.
- ~ The scope must be unambiguous and certain. A tiny prototype is performed to realize the object of desire.

(iii) The Project

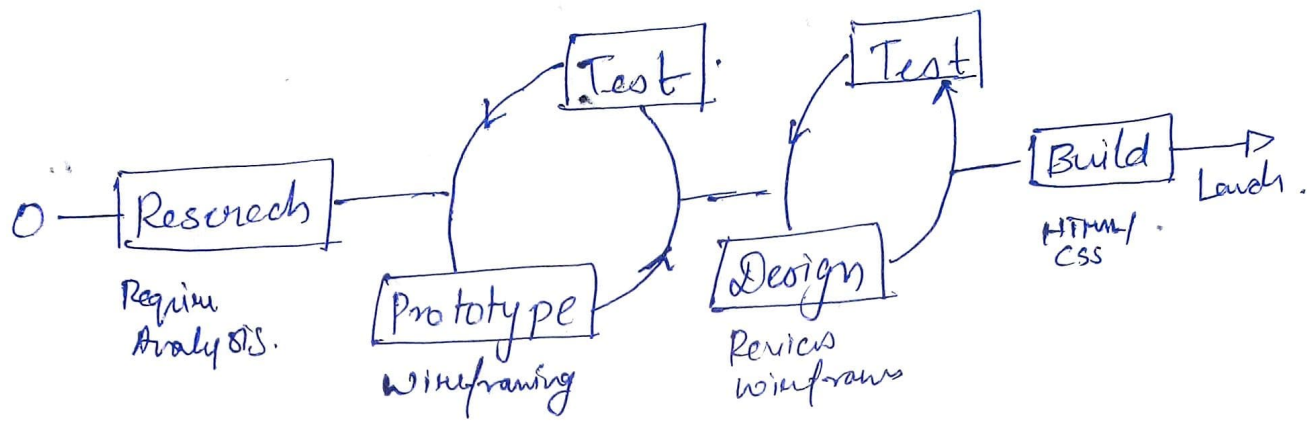
- ~ Here the most important thing is to select an appropriate optimized model.
- ~ There are various models like, waterfall, iterative, evolutionary, RAD etc.
- ~ In practice we can use the best suited model or combinations to proceed.

(iv) The Project

- ~ To manage a successful project we must be quick at understanding what can go wrong.
- ~ It is a series of steps that can make the right decision for a successful project.

A8.2) UI design Process

The figure can depict a general way of UI design.



- ~ UI - stands for User Interface
- ~ It is the part of the software ~~that~~ which is visible to the user.
- ~ This ~~helps~~ is one of the most important part as visual appeal is of top priority for any user.

1. Research

- ~ This begins with the analysis of the required
- ~ We ~~are~~ need to understand in which field we would be deploying the application.
- The UI for a banking system is entirely different from an UI for a media player.

2. Prototype & Testing

- ~ This will be a rough prototype usually drawn using hands
- ~ This will give the first visual aid to the client - how the product will look like
- ~ This goes through a series of revision until a satisfactory design is realized.

3. Design & Testing

- ~ In this phase computer aided software tools are used to literally design how ~~out~~ our web product will look like.
- ~ Selection of color-pallets, curves, screen, transitions etc. are of much importance.
- ~ Eg:- Wire framing in figma, Adobe XD, Sketch etc.

4.) Build

- ~ Now it is time to bring the design to life.
- ~ This is done using coding softwares like HTML, CSS or even ~~DATA~~ ~~such that~~
- ~ This will make the design responsive and ~~are~~ ready for deployment.

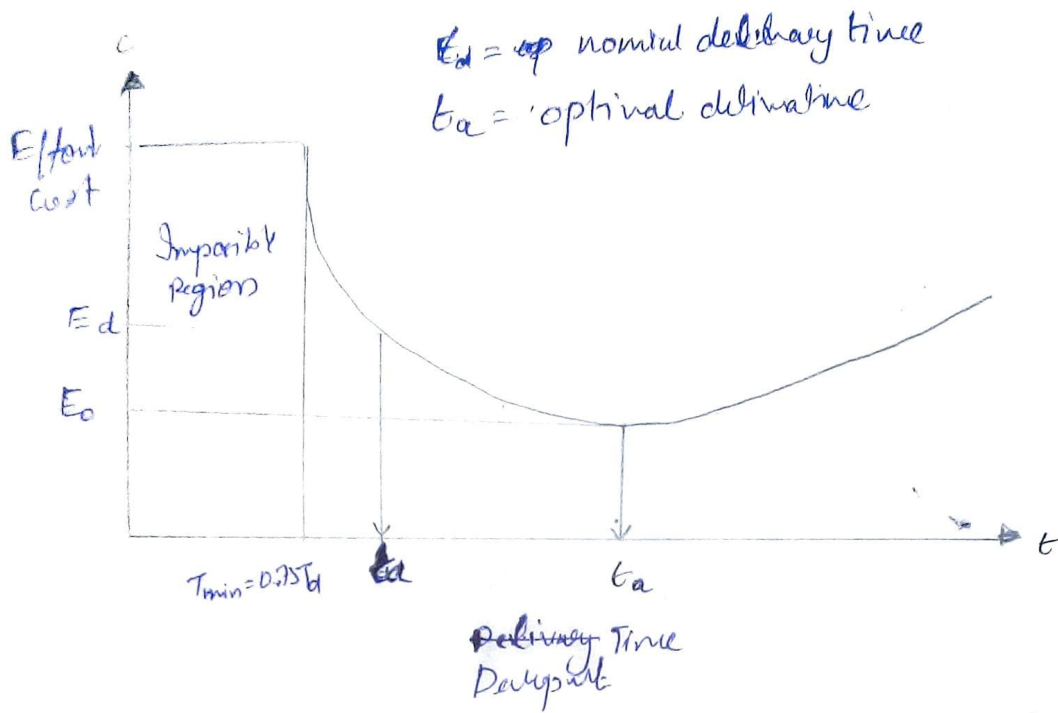
A9.) Relation b/w Effort Applied & Delivery Time

Effort

- ~ Effort is the amount of work or number of work units used to complete an activity.
- ~ The effort is the number of hours worker spend focused on a particular task.
- ~ Effort is most often expressed in staff hours, days or weeks.
- ~ Stakeholders often want to know how much the project will cost.
- ~ This is chiefly depends upon the measure of time members spend on the project.

PNR

- ~ Research by Putnam Mordan found that for projects that require communication and learning the efforts follow a Rayleigh distribution.
- The Putnam Mordan Rayleigh Curve is shown below:



- ~ This graph shows that there is a non-linear ~~dist~~ relation between Effort and Delivery Time.
- ~ If ~~from~~ we move from right to left we see that the curve gets steeper and steeper (indicating increased cost) but not much of a shortening in time line.
- ~ Most companies are looking for the best compromise between low cost and short time lines.
- ~ This is indicated by the nominal delivery time ' t_d '.

A10. SCM - Software Configuration Management

- Software Configuration management is the process of controlling and monitoring changes to products.

~ It helps to systematically manage and organize, and control the change in documents codes and other entities during the Software Development cycle.

~ It involves the following processes:

1. Identification & Establishment:

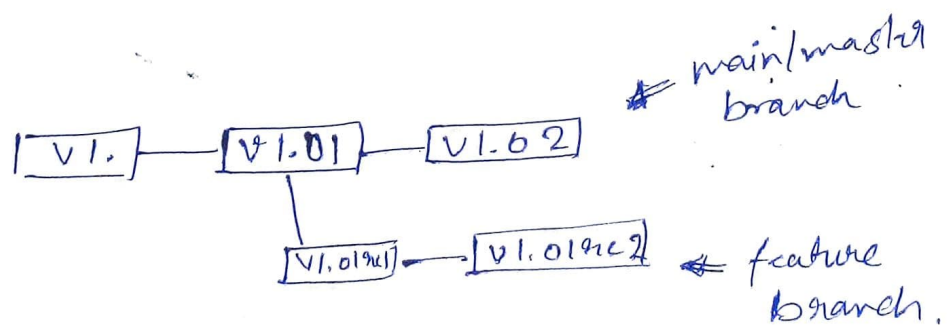
~ Identifying the configuration items from products that compose baselines at given points in time.

~ Establishing relationship among item & by creating mechanisms (like GitHub Actions) to manage multiple levels of control & procedure.

2. Version Control

~ This is the process of version the software changes at each and every stage.

~ Some of the common VCS Tools are git, mercurial and SVN.



3.) Change Control

- ~ A change request is submitted to the manager.
- ~ It is evaluated and presented as a change report issued by change control board.
- ~ It has to deal with the people formally entering and leaving the during the software development.
- ~ On the basis of software it is majorly related to databases.

4.) Configuration auditing

- ~ This audit focus upon the technical correctness of the configuration object that has been modified.
- ~ It is a formal technical review.
- ~ The audit confirms or rejects the software based on the inference of the review upon the enlighten points.

5.) Reporting

- ~ It provides up to date, status, configuration data to developer, testis end users, customers and stakeholders.
- ~ Various documents are released Admin guides, Client FAQ, Release notes etc.