H-K Information Complexity

$$H \cdot K_i = (Wt \cdot factor) * (in_i * out_i)^2$$

i = i-th phase / module.

Module ≠	a	b	c	d	e
in	4	3	4	5	6
out	1	2	5	2	4

Wt · factor, a = 1·0, b = 0·5, c = 2·0, d = 2·5 & e = 3·0

Gini index

$a = 1.0 * (4 \times 1)^2$

$b = 0.5 * (3 \times 2)$

$$HK = \sum_{i=1}^{n} Wi * (i_i O_i)^2$$

$HK_a = 1.0 * (4 \times 1)^2 = 1.0 * 16 = 16$

$HK_b = 0.5 * (3 \times 2)^2 = 0.5 * 36 = 18.0$

$HK_c = 2.0 \times (4 \times 5)^2 = 2.0 * 400 = 800$

$HK_d = 2.5 \times (5 \times 2)^2 = 2.5 * 100 = 250$

$HK_e = 3.0 \times (6 \times 4)^2 = 3.0 \times 576 = 1728$

$\therefore HK = 16 + 18.0 + 800 + 250 + 1728 = 2812$

McCabe's Cyclomatic Complexity

$$V(G) = E - N + 2$$
$$= d + 1$$
$$= r + 1$$

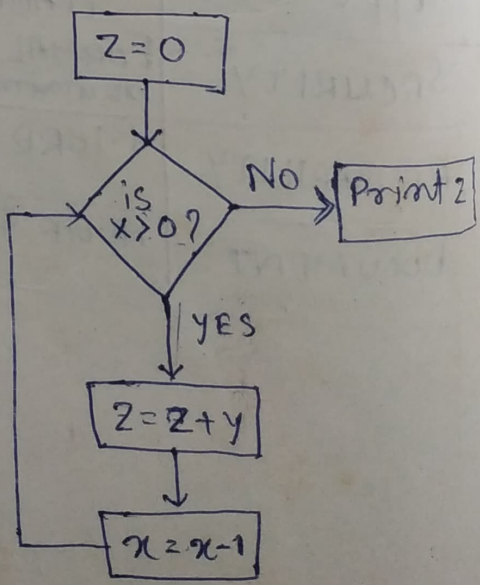
This complexity is the mejorment of validity of code designer. V(G) stands for complexity of graph.

E = Edge
N = Node or Vertices
d = decision nodes
r = close regions/loop.

Prob-1) WACP to generate ~~febinice~~ fibonacci no. till 20 and apply cyclometic complexity of ~~mejorment~~ measurement

```
Z = 0
While x > 0
Z = Z + y
x = x - 1
end while,
Print (Z)
```

Flowchart



$E = 5$

$N = 5$

$d = 1$

$r = 1$

$v(G) = E - N + 2$

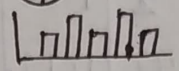
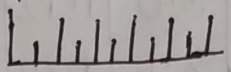
$= 5 - 5 + 2 = 2$

$= d + 1 \quad = 1 + 1 = 2$

$= r + 1 = 1 + 1 = 2$

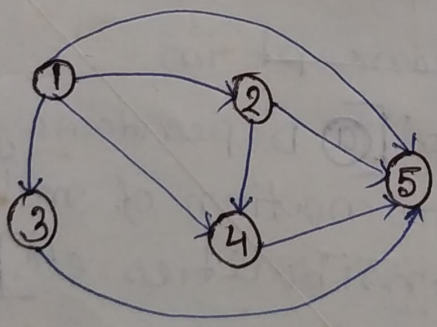
PRINCIPLE OF SCHEDULING: A no. of basic priciple guide s/w project scheduling though are also consider as features of project scheduling.

TOOLS USING SCHEDULING:

1. Graant Chart.
2. Flow chart
3. Pie chart → 
4. Bar Graph → 
5. Line Graph → 
6. Pareto Diagram.
7. Histogram Scheduling.
8. PERT_CPM

PERT - Project Evalution and Review Technique.

CPM - Critical Path Method.



PERT

CPM

1. 1-5 → 3.863
2. 1-2-5 → (1-2) + (12.166) → 14.166
3. 1-2-4-5 → (1-2) + (2-4) + (4-5) → 12.1
4. 1-3-5 → (1-3) + (3-5) → 6.16
5. 1-4-5 → (1-4) + (4-5) → 5.167

TIME

Activity	To	Tm	Tp	te (to + 4tm + tp / 6)
1·2	1	2	3	2
1·3	3	4	5	4
1·4	1	2	4	2·166
1·5	2	4	5	3·833
2·4	6	7	8	7
2·5	10	12	15	12·166
3·5	1	2	4	2·16
4·5	2	3	5	3·1

to = Optimetic / minimum time

tm = most likely.

tp = pasimistic / maximum

te = Estimated time.

Date - 19/10/16

Feature	Throwaway Prototyp-ing.	Evolutionary prototyping
1. Develope-ment approach.	1. Quick & Dirty	1. Rigorous (Structure) [Get a template]
2. What to build.	2. Build only difficult and critical parts.	2. Build understood parts and build solid foundations on that.
3. Design	3. Optimise development types.	3. Optimise the modifica-tion & quality.
4. Goal	4. Throw it away.	4. ~~Evault~~ Evolve it.

VERIFICATION	VALIDATION
i) It is a static process to verify documents, design and code.	i) It is a dynamic process to test the actual products.
ii) It does not involve executing code.	ii) It involves executing code.
iii) Verification comes before validation.	iii) It generally follows verification.
iv) It ans. the questions that are we building the product right?	iv) It ans the questions that are we building the right product?

Testing Principles :

1) Effective testing, not ~~exustive~~ exhaustive testing.

2) Testing is not a single phase performed in SDLC.

3) Early testing is the best policy.

4) Destructive approach for constructive testing.

5) Probablity of existence of an error in a section of a program is proportional to the no. of errors already found in that sect

6) Testing strategy should start at the smallest module level and expand towards the whole program.

7) Testing should also be performed by an independent team.

8) Everything must be recorded in SW testing

9) Invalid i/Ps & unexpected behaviour have a high probablity of finding an error.

10) Tester must participate in specification and design review.

Difference B/W BLACK BOX TESTING & WHITE BOX TESTING:

BBT	WBT
It is also called functional or behavioral testing.	It is also called structural testing.
It is known as gray box testing.	It is known as glass box testing.
It examines some fundamental aspects of the system with little regard for internal logical structure of the system.	Here the procedural details, all the logical paths, all the internal data structures are closely examined.
During the types of testing the program cannot be tested 100%.	It tests the program ~~throw~~ thoroughly.
It is suited for learge projects.	It is suitable for small projects.