

ehindistudy.com

# computer science notes in hindi

PDF generated December 17, 2018 by Kalin's PDF Creation Station WordPress plugin

## Agile model in hindi एजाइल मॉडल क्या है?

by yugal joshi - Wednesday, September 05, 2018

<https://ehindistudy.com/2018/09/05/agile-model-kya-hai-hindi/>

### Agile model in hindi

Agile model जो है वह iterative तथा incremental models का एक combination है अर्थात् यह iterative तथा incremental models से मिलकर बना है.

Agile model में process adaptability तथा customer satisfaction पर ध्यान दिया जाता है.

पहले के समय में iterative [waterfall model](#) का प्रयोग सॉफ्टवेयर को बनाने में किया जाता था. परन्तु आज के समय में developers को बहुत सी परेशानियों का सामना करना पड़ता है. सबसे बड़ी परेशानी यह है कि software development के बीच में customer सॉफ्टवेयर में changes करने को कहता है. इन changes को करने में बहुत सारा time तथा money लगता है.

तो इन सब कमीयों को पूरा करने के लिए 1990 के दशक में agile model को प्रस्तावित किया गया.

agile model को मुख्यतया software development के बीच में changes करने के लिए ही बनाया गया था जिससे कि सॉफ्टवेयर प्रोजेक्ट को जल्दी से पूरा किया जा सकें.

agile model में निम्नलिखित steps होते हैं:-

- 1:- [requirement gathering](#)
- 2:- requirement analysis
- 3:- design
- 4:- coding
- 5:- [unit testing](#)
- 6:- acceptance testing

agile model में software product को छोटे incremental भागों (parts) में बाँट दिया जाता है. इसमें सबसे पहले सबसे छोटे part को विकसित किया जाता है और उसके बाद उससे बड़ा.

और प्रत्येक incremental part को iteration पर विकसित किया जाता है.

प्रत्येक iteration को छोटा रखा जाता है जिससे कि उसे आसानी से manage किया जा सके. तथा उसे दो तीन हफ्तों में पूरा किया जा सके. एक समय में केवल एक ही iteration को plan, develop तथा deploy किया जाता है.

## **principles of Agile model in hindi**

इसके principles निम्नलिखित हैं:-

1:- software development के समय customer के साथ contact बनाये रखने के लिए तथा requirement को समझने के लिए development team में एक customer representative होता है. जब किसी iteration को पूरा कर लिया जाता है तो stakeholders तथा customer representative उसे review करते हैं तथा requirements को दूबारा evaluate करते हैं.

2:- customer की requirements को समझने के लिए working software का demo दिया जाता है. अर्थात् इसमें सिर्फ documentation पर depend नहीं रहते हैं.

3:- सॉफ्टवेयर के incremental versions को कुछ हफ्तों बाद customer representative को deliver करना पड़ता है.

4:- इस model में यह सलाह दी जाती है कि development team का size छोटा (5 से 9 लोग) होना चाहिए जिससे कि team member जो है वह face to face communicate कर सकें.

5:- agile model इस बात पर focus करता है कि सॉफ्टवेयर में जब भी कोई changes करने हो तो उसे जल्दी से पूरा कर लिया जाए.

6:- agile development में दो programmers एक साथ काम करते हैं. एक programmer कोडिंग करता है तो दूसरा उस code को review करता है. दोनों programmers अपने कामों को बदलते रहते हैं यानी कि कभी कोई coding करता है तो कभी कोई review.

## **advantage of agile model in hindi**

इसके लाभ निम्नलिखित हैं:-

1:- इसमें दो programmers एक साथ काम करते हैं जिससे coding बहुत ही अच्छी होती है तथा उसमें बहुत ही कम गलतियाँ होती हैं.

2:- इसमें software project को बहुत कम समय में पूरा कर लिया जाता है.

3:- इसमें customer representative को प्रत्येक iteration का idea होता है जिससे वह requirement को आसानी से change कर सकते हैं.

4:- यह सॉफ्टवेयर development की बहुत ही वास्तविक approach है.

5:- इसमें teamwork पर ध्यान दिया जाता है.

6:- इसमें rules बहुत कम होते हैं और documentation भी ना के बराबर होता है.

7:- इसमें planning की जरूरत नहीं पड़ती है.

8:- इसे आसानी से manage किया जा सकता है.

9:- यह developers को flexibility प्रदान करता है.

## **disadvantage of agile model in hindi**

इसकी हानियाँ निम्नलिखित हैं

1:- यह complex dependencies को handle नहीं कर पाता है.

2:- इसमें formal documentation की कमी की वजह से development में confusion हो जाता है.

3:- यह ज्यादातर customer representative पर निर्भर रहता है अगर customer representative कोई गलत information दे देता है तो सॉफ्टवेयर गलत बन सकता है.

4:- इसमें केवल experienced programmers ही कोई decision ले सकते हैं. नए programmers कोई भी decision नहीं ले सकते हैं.

5:- इसमें software development के शुरुवात में सॉफ्टवेयर को बनाने में कितना effort तथा time लगेगा इसका पता नहीं लग पाता है.

## **agile SDLC models**

agile में निम्नलिखित models होते हैं:-

- Scrum
- crystal methodologies
- DSDM
- feature driven development (FDD)
- lean software development
- extreme programming (XP)

निवेदन:- अगर आपको यह पोस्ट पसंद आई हो तो इसे अपने दोस्तों के साथ अवश्य share करें.

---

# alpha & beta testing in hindi?

by yugal joshi - Monday, August 08, 2016

<https://ehindistudy.com/2016/08/08/alpha-beta-testing-hindi/>

## Alpha testing in hindi:-

alpha टेस्टिंग को निम्नलिखित बिन्दुओं के आधार पर आसानी से समझ से सकते हैं:-

\*यह टेस्टिंग acceptance टेस्टिंग का एक प्रकार है.

\*customers के लिए सॉफ्टवेर प्रोडक्ट को release करने से पहले उसमें संभावित defects तथा bugs को detect करना इस टेस्टिंग का मुख्य उद्देश्य होता है.

\*इस टेस्टिंग को हमेशा developers के द्वारा developer site(जहाँ engineers काम करते हैं) पर ही परफॉर्म किया जाता है.

\*यह टेस्टिंग तब की जाती है जब सॉफ्टवेर बनकर तैयार हो जाता है अगर alpha टेस्टिंग के बाद defects के कारण कुछ बदलाव करने होते हैं वह कर लिए जाते हैं.

\*यह टेस्टिंग दो steps में पूर्ण होती है, पहले स्टेप में इस टेस्टिंग को developers के द्वारा परफॉर्म किया जाता है जबकि दुसरे स्टेप को software quality assurance(SQA) टीम के द्वारा परफॉर्म किया जाता है.

\*यह टेस्टिंग पब्लिक के लिए बंद होती है अर्थात् इस टेस्टिंग में पब्लिक का कोई involvement नहीं होता है.

\*यह टेस्टिंग हमेशा virtual environment में ही की जाती है.

What is  
Alpha  
&  
BETA  
Testing in Hindi

### **Beta testing in hindi:-**

beta testing को निम्नलिखित बिन्दुओं के आधार पर आसानी से समझ सकते हैं.

\*यह भी एक acceptance टेस्टिंग का एक प्रकार है तथा इसे field टेस्टिंग भी कहते हैं.

\*इस टेस्टिंग को end users के द्वारा real environment में परफॉर्म किया जाता है.

\*इसे टेस्टिंग टीम के द्वारा परफॉर्म नहीं किया जाता है.

\*beta टेस्टिंग का उद्देश्य users के perspective के आधार पर सॉफ्टवेर में उपस्थित दोष तथा समस्या को detect करना है तथा users से उपलब्ध फीडबैक को प्राप्त करना है, ताकि सॉफ्टवेर के अगले updated version में इन समस्याओं को दूर किया जा सकें.

\*यह टेस्टिंग real time environment में की जाती है.

\*यह टेस्टिंग पब्लिक के द्वारा की जाती है.

\*alpha टेस्टिंग के बाद beta टेस्टिंग की जाती है और beta टेस्टिंग के बाद कोई और टेस्टिंग नहीं होती है.

- आपको यह पोस्ट कैसी लगी हमें comment के माध्यम से बताये तथा अपने दोस्तों के साथ share करें. Alpha तथा Beta टेस्टिंग के बारे आप अगर कुछ अन्य बिंदु जानते हो तो हमें बताएं. धन्यवाद.

---

## Black box testing & White box testing in hindi

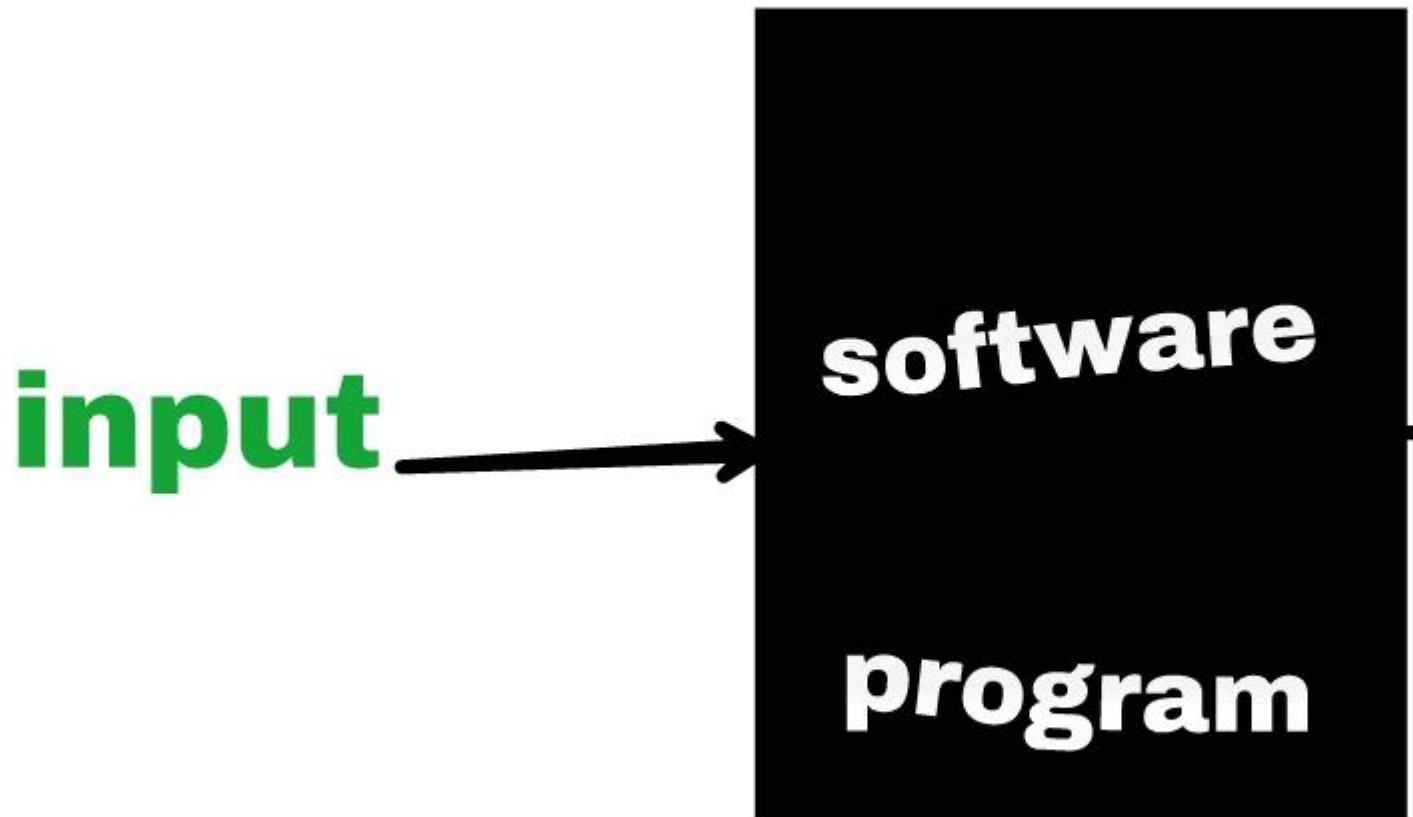
by yugal joshi - Friday, May 27, 2016

<https://ehindistudy.com/2016/05/27/black-box-testing-white-box-testing-in-hindi-2/>

### Black box testing in hindi:-

black box वह टेस्टिंग होती है जिसमें tester को सॉफ्टवेयर के अंदर के structure का ज्ञान होना जरूरी नहीं होता है तथा टेस्टर को coding की knowledge होना भी जरूरी नहीं होता है।

# black box



Black box टेस्टिंग एक ऐसी टेस्टिंग तकनीक है जिसमें परीक्षण के तहत सॉफ्टवेयर की कार्यक्षमता को टेस्ट किया जाता है जबकि सॉफ्टवेयर के internal कोड स्ट्रक्चर को नहीं देखा जाता है।

Black box टेस्टिंग में सॉफ्टवेयर के सिर्फ इनपुट तथा आउटपुट पर ध्यान दिया जाता है।

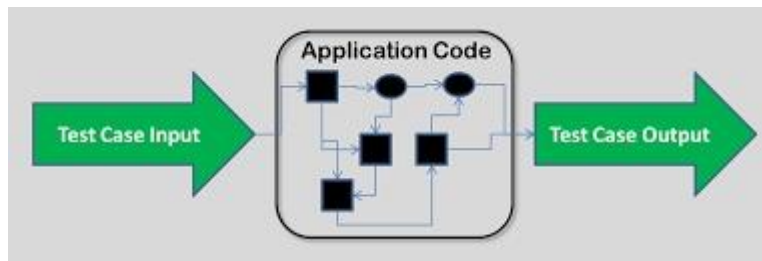
**उदाहरण के लिए:-** माना हमने whats app के सॉफ्टवेयर की black box टेस्टिंग करनी है तो हम उसके केवल इनपुट तथा आउटपुट पर ध्यान देते हैं जबकि हमें उसके internal structure की कोडिंग का ज्ञान नहीं होता है।



Black box टेस्टिंग को इनपुट-आउटपुट टेस्टिंग भी कहा जाता है। इस टेस्टिंग को black box टेस्टिंग इसलिए कहते हैं क्योंकि जो सॉफ्टवेयर होता है वह tester के लिए black box की तरह होता है जिसके अंदर वह नहीं देख सकता।

## White Box Testing in hindi:-

वह टेस्टिंग जिसमें टेस्टर को सॉफ्टवेयर के internal स्ट्रक्चर का ज्ञान होता है white box टेस्टिंग कहलाती है। इस प्रकार की टेस्टिंग सॉफ्टवेयर के अंदर की working पर आधारित होती है।



इस टेस्टिंग को white box इसलिए कहते हैं क्योंकि जो सॉफ्टवेयर होता है वह टेस्टर के लिए white box की तरह होता है जिसके अंदर वह देख सकता है।

इस प्रकार की टेस्टिंग में टेस्टर को प्रोग्रामिंग का ज्ञान होता है। इस टेस्टिंग का बड़ा फायदा यह है कि हम सॉफ्टवेयर डेवलपमेंट के शुरुआत की stage में ही त्रुटियों का पता लगा सकते हैं।

इस टेस्टिंग को open box

testing , glass box testing तथा structural टेस्टिंग भी कहते हैं।

निवेदन:-अगर आपके पास भी कोई हिंदी में topic या किसी subject से सम्बंधित नोट्स हैं तो कृपया उसे अपनी फोटो के साथ E-mail करें. हमारा e-mail:- ehindistudy@gmail.com है। हम उसे यहां आपकी फोटो के साथ प्रकाशित करेंगे।

---

## coupling and cohesion in hindi

by yugal joshi - Tuesday, May 15, 2018

<https://ehindistudy.com/2018/05/15/coupling-cohesion-hindi/>

## **coupling and cohesion in hindi:-**

अगर software का design अच्छा होता है तो [software quality](#) बेहतर होती है तथा software design को analyze करने के बहुत सारे तरीके होते हैं जैसे-coupling, cohesion, factoring, system shape आदि.

### **Cohesion-**

cohesion एक measurement (माप) है जो कि एक module के अंदर elements की intra-dependability की degree को डिफाइन करता है.

अर्थात cohesion में यह देखा जाता है कि एक module के elements एक दूसरे से कितनी करीब से सम्बंधित हैं. अर्थात modules के elements की relationship एक दूसरे से कितनी गहरी है.

जितना ज्यादा cohesion होगा उतना ही अच्छा software design होगा.

Module का cohesion जितना अधिक होगा, उतना ही कम module के मध्य coupling होगी.

### **Types of cohesion in hindi-**

Cohesion निम्नलिखित प्रकार का होता है.

- 1- coincidental
- 2- logical
- 3- temporal
- 4- procedural
- 5- communicational
- 6- sequential
- 7- functional

# Cohesion

- In order from good (high) to bad (low)

- Informational
- Functional
- Communicational
- Procedural
- Temporal
- Logical
- Coincidental

Good



Bad

**1- coincidental cohesion-** यह cohesion सबसे निम्न स्तर का cohesion होता है. यह तब घटित होता है जब module के elements के मध्य कोई relationship नहीं होती. यह तब होता है जब प्रोग्राम को छोटे modules में तोड़ दिया जाता है.

**2- logical cohesion-** एक मॉड्यूल में logical cohesion तब होता है जब एक मॉड्यूल के elements के मध्य लॉजिकल relationship होती है. तथा elements एक ही लॉजिकल class में रखे गए functions को परफॉर्म करते हैं.

**3- temporal cohesion-** यह cohesion जो है वह logical cohesion की तरह होता है परन्तु इसमें elements एक ही समय पर execute होने चाहिए. temporal cohesion लॉजिकल cohesion से उच्च स्तर का होता है क्योंकि इसमें elements एक ही समय पर प्रोसेस होते हैं.

**4- procedural cohesion-** जब मोड्यूल के elements एक साथ समूह में होते हैं तो वह एक task को पूरा करने के लिए एक sequence में execute होते हैं इसी को procedural cohesion कहते हैं.

**5- communicational cohesion-** जब module के elements एक साथ group में होते हैं जो कि एक कार्य को पूरा करने के लिए sequence में execute होते हैं तथा ये सभी elements एक ही समय डेटा (इनफार्मेशन) पर कार्य करती है तब इसे communicational cohesion कहते हैं

**6- sequential cohesion-** जब module के elements एक साथ एक group में होते हैं क्योंकि एक element का output दूसरे element के लिए input की तरह होता है तो हमें sequential cohesion प्राप्त होता है.

**7- functional cohesion-** यह cohesion का सबसे उच्चतम स्तर होता है module के सभी elements एक function को परफॉर्म करने से सम्बन्धित होते हैं.

## Coupling-

coupling एक measurement (माप) है जो एक प्रोग्राम के modules के मध्य inter-dependability के स्तर को डिफाइन करता है.

अर्थात यह एक प्रकार की measurement है जो कि यह डिफाइन करती है कि software के components एक दूसरे पर किस प्रकार निर्भर(depend) तथा interact करते हैं.

coupling जितनी कम होगी software design उतना ही बेहतर होगा.

## Types of coupling in Hindi-

Coupling निम्नलिखित प्रकार की होती है:

1- content कपलिंग

2- common कपलिंग

3- external कपलिंग

4- control कपलिंग

5- stamp कपलिंग

6- data कपलिंग

## Coupling

- In order from good (low) to bad (high)

- Data Coupling
- Stamp Coupling
- Control Coupling
- Common Coupling
- Content Coupling

Good



Bad

**1- Content Coupling-** content कपलिंग जो है वह सबसे उच्च स्तर का कपलिंग होता है यह तब होता है जब कोई module किसी दूसरे module की आंतरिक क्रियाविधि पर निर्भर होता है. इसका मतलब यह है कि दूसरे module में कोई बदलाव होगा तो उस dependent module में भी बदलाव होगा.

**2- common coupling-** इसे global कपलिंग भी कहते हैं यह तब घटित होता है जब समान global data दो modules के द्वारा share किया जाता है. इसमें दोनों modules में बदलाव आएगा जब global data में बदलाव किये जायेंगे तो.

**3- external coupling-** इस प्रकार की कपलिंग तब घटित होती है जब external data format तथा communication protocol को दो modules के द्वारा share किया जाता है.

**4- control coupling-** इस प्रकार की कपलिंग में, एक module दूसरे module के flow को नियंत्रित करता है तथा डेटा को एक से दूसरे module में भेजता है.

**5- stamp coupling-** इस प्रकार की कपलिंग में बहुत सारे modules समान प्रकार के [डेटा स्ट्रक्चर](#) को share करते हैं तथा इसके अलग-अलग भागों में कार्य करते हैं.

**6- data coupling-** दो modules के मध्य data coupling तब होगी जब डेटा केवल उनके ही मध्य pass होगा.

**निवेदन:-** अगर आपको यह पोस्ट अच्छी लगी हो तो मुझे comment के द्वारा बताइए तथा इस पोस्ट को share करें.

---

## Feasibility study in hindi

by yugal joshi - Wednesday, June 01, 2016

<https://ehindistudy.com/2016/06/01/feasibility-study-in-hindi/>

### Feasibility Study in hindi:-

किसी भी सिस्टम की सफलता के लिए सिस्टम की क्षमता का परीक्षण करना Feasibility Study कहलाती है।

Feasibility study का मुख्य उद्देश्य यह निर्धारित करना है कि सिस्टम को विकसित करना financially तथा technically रूप से संभव है या नहीं।

Feasibility study का उद्देश्य problem को solve करना नहीं होता है बल्कि यह निर्धारित करना है कि प्रॉब्लम को solve किया जा सकता है या नहीं।

इसमें सिस्टम में आने वाली सभी दिक्कतों को analyze किया जाता है, इसमें सिस्टम की जानकारी को गहराई पूर्वक जाँचा जाता है और यह निर्धारित किया जाता है कि सिस्टम को विकसित करने में किस प्रकार से सफलता पायी जा सकती है।

**types of feasibility study in hindi:-**

Feasibility study निम्नलिखित तीन प्रकार की होती है:-

- 1:-Technical Feasibility
- 2:-Operational Feasibility
- 3:-Economical Feasibility

# feasibility study

```
graph TD; A[feasibility study] --> B[Technical]; A --> C[operational]; A --> D[Economic]
```

**Technical**

**Economic**

**operational**



**1:-Technical Feasibility:-** इस feasibility में सिस्टम की टेक्निकल जरूरतों को निर्धारित किया जाता है। इसमें यह देखा जाता है कि जो प्रस्तावित सिस्टम है उसके लिए जो टेक्नोलॉजी चाहिए वह उपलब्ध हो तथा उस टेक्नोलॉजी को सिस्टम में किस प्रकार integrate किया जायें। नयी टेक्नोलॉजी में आने वाली सभी प्रकार की complexities को handle करने के लिए तकनीकी रूप से सक्षम एक्सपर्ट की जरूरत भी होती है।

**2:-Operational Feasibility:-** इस feasibility में यह निर्धारित किया जाता है कि एक प्रस्तावित सिस्टम किस प्रकार problems को solve करेगा तथा सिस्टम में किस प्रकार के बदलाव आये हैं? इसमें यह देखा जाता है कि जो सिस्टम है क्या वह यूज़र्स के लिए लाभकारी है या नहीं तथा क्या वह जरूरतों को पूरा करने में सक्षम है या नहीं।

**3:-Economical Feasibility:-** इस feasibility में यह निर्धारित किया जाता है कि जो प्रस्तावित सिस्टम है उसमें कितना खर्चा आएगा तथा उसमें कितना benefit मिलेगा?

Economic feasibility को cost benefit analysis भी कहते हैं।

ENGLISH? इसकी किसे जरूरत है? मैं कभी इंग्लैंड नहीं गया।

---

## Importance of Software testing in hindi

by yugal joshi - Wednesday, August 03, 2016

<https://ehindistudy.com/2016/08/03/importance-software-testing-hindi/>

### importance of software testing in hindi:-

इस पोस्ट में हम सॉफ्टवेर टेस्टिंग की importance या आवश्यकता जो है उसके बारे में पढ़ेंगे.

सॉफ्टवेर को बनाते समय हम से कुछ न कुछ गलती हो जाती है क्योंकि हम इंसान हैं और इंसान लगातार गलतियाँ करता ही है. कुछ गलतियाँ हानिकारक नहीं होती हैं तो कुछ गलतियाँ बहुत ही ज्यादा खतरनाक हो सकती हैं.

इन्हीं सब गलतियों को सुधारने के लिए हमें सॉफ्टवेर को test करना पड़ता है.

तो आइये जानते है सॉफ्टवेर टेस्टिंग की आवश्यकता क्यों पडती है?

## importance of software testing in hindi

- 1:-सॉफ्टवेर में सम्भावित सभी defects तथा errors को ढूढ़ने के लिए सॉफ्टवेर टेस्टिंग की जाती है.
- 2:-सॉफ्टवेर प्रोडक्ट की क्वालिटी को सुनिश्चित करने के लिए भी सॉफ्टवेर टेस्टिंग की जाती है. अगर क्वालिटी बेहतर होगी तो customer अधिक संख्या में सॉफ्टवेर प्रोडक्ट का इस्तेमाल करेंगे.
- 3:-सॉफ्टवेर की performance को बढ़ाने के लिए.
- 4:-यह सिद्ध करने के लिए कि सॉफ्टवेर में कोई fault(गलती) नहीं है.
- 5:-सॉफ्टवेर की reliability को बढ़ाने के लिए.
- 6:-यह सुनिश्चित करने के लिए सॉफ्टवेर customer की आवश्यकतानुसार बना है या नहीं?
- 7:-बिज़नस में बने रहने के लिए भी सॉफ्टवेर टेस्टिंग की जाती है.

---

## McCabe's Complexity in hindi

by yugal joshi - Wednesday, June 15, 2016

<https://ehindistudy.com/2016/06/15/mccabes-complexity-in-hindi-2/>

## McCabe's Complexity in hindi:-

McCabe की complexity को cyclomatic complexity कहते हैं। यह एक सॉफ्टवेयर metric है इस complexity का प्रयोग सॉफ्टवेयर प्रोग्राम की complexity को ज्ञात करने के लिए किया जाता है।

Cyclomatic complexity प्रोग्राम के source कोड के द्वारा सीधे ही independent paths की संख्या को measure करती है। independent paths की संख्या जितनी अधिक होगी code उतना ही काम्प्लेक्स होगा।

Independent paths वह path है जिसका कम से कम एक edge होता है तथा जिसका पहले traversal नहीं हुआ होता है।

इस complexity को 1976 में Thomas McCabe ने प्रस्तावित किया था।

इसको गणितीय रूप में निम्नलिखित प्रकार से समझा जा सकता है:-

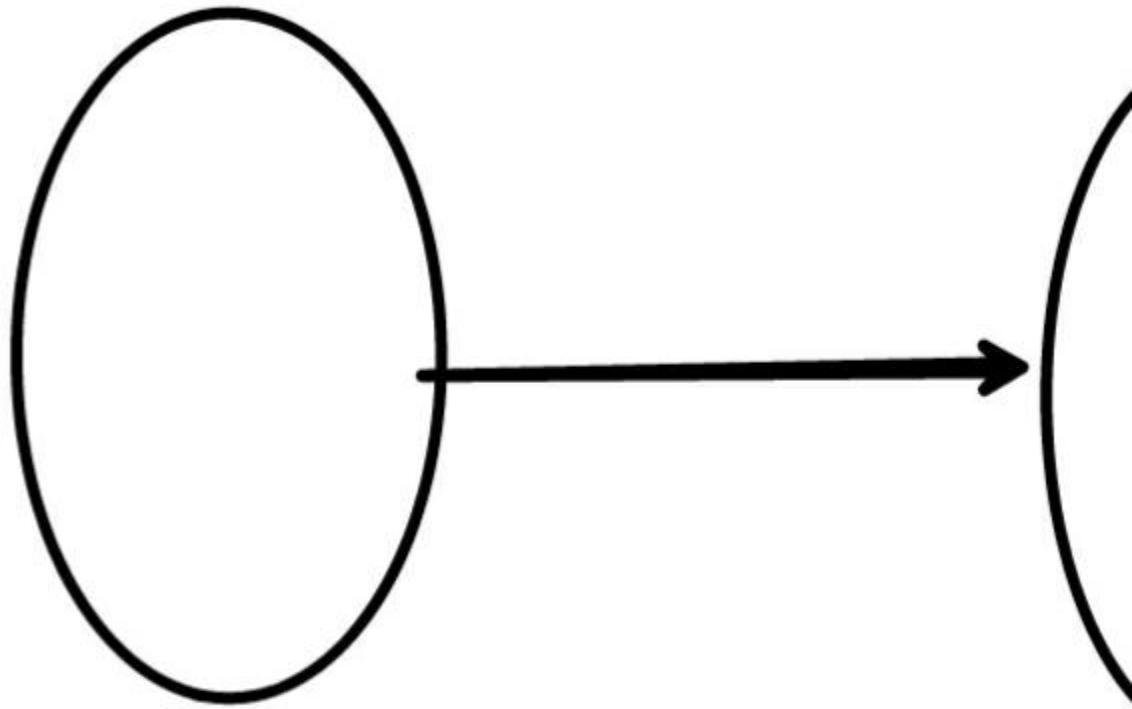
$[McCabe's\ complexity\ V(G) = E - N + 2]$

जहां;

E= edges की संख्या

N=Vertices की संख्या

P=disconnected paths की संख्या है।



$$e=1, p=1, n=2$$

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

Fig:-ex of cyclomatic complexity

## properties of McCabe's complexity:-

इसके निम्नलिखित गुण होते हैं:-

- 1:-  $V(G) \geq 1$
- 2:- ग्राफ में  $V(G)$  independent path की अधिकतम संख्या है।
- 3:- यदि  $V(G)=1$  तो ग्राफ में केवल एक path होगा।
- 4:- यह complexity को 10 से कम कर देता है।

अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं।

---

# requirement elicitation (gathering) in hindi and requirement elicitation techniques in hindi

by yugal joshi - Tuesday, August 14, 2018

<https://ehindistudy.com/2018/08/14/requirement-elicitation-hindi/>

## REQUIREMENT ELICITATION in hindi

Elicitation का अर्थ collect करना होता है और requirement Elicitation का मतलब requirement को collect करना होता है।

Requirement elicitation एक ऐसी प्रोसेस है जिसमें सॉफ्टवेयर के requirements को collect करने के लिए customers, stake holders तथा end users के साथ interact किया जाता है.

requirement elicitation को कभी कभी requirement gathering भी कहते हैं.

किसी भी software को बनाने के लिए सबसे पहले requirements को जानने व समझने की आवश्यकता होती है। और इन सभी requirements को collect करने के लिए एक medium (माध्यम) की आवश्यकता होती है जो software से related सभी आवश्यक जानकारी को collect करते हैं। ये medium stack-holder या end-user होते हैं। सभी आवश्यक requirements developer को end-user या stack-holder द्वारा दी जाती हैं।

Stockholder एक individual person होते हैं जो system या organization से direct या indirect कनेक्ट होते हैं। जैसे :- manager , tester , software engineer , developer आदि.

इन end-user या stack-holder द्वारा दी गयी requirement को collect करके software बनाया जाता है। software development process तभी सफल हो सकता है जब customer और developer के बीच प्रभावशाली partnership हो। requirement elicitation technique की सफलता customer , developer , user और analyst के समझदारी के ऊपर depend करता है।

## **Problem arises during requirement gathering (requirement को इक्कठा करने समय आने वाली परेशानियाँ)**

**Problem of scope :-** जब software बना रहे होते हैं तो उसकी boundary को mention किया जाता है। एक software की fix boundary है या नहीं उसे mention किया जाता है। एक software की boundary fix होनी चाइये यदि software की boundary fix नहीं होती है तो problem create हो जाती है।

उदाहरण के लिए :- software बनने के लिए cost व budget कितना होगा , developer की salary कितनी होगी यह सब mention किया जाता है।

Software को बनते समय यदि scope को define नहीं किया गया तो problem create हो जाती है। इसलिए software बनते समय scope को पूरी तरह से define करना चाइये और सभी information mention करनी चाइये।

**Problem of understanding: -** software से संबंधित सभी modules के बारे में पूरी जानकारी होनी चाइये तभी software से संबंधित information collect कर सकते हैं। यदि software के सभी modules को अच्छे से समझ नहीं पाएंगे तो information भी collect नहीं कर सकते।

**Problem of volatility (अस्थायी) :-** environment तथा user के जरिये मिली जानकारी fix नहीं होती है। किसी भी software को बनने के लिए उसका scope fix होना चाइये यदि scope change होगा तो information collect नहीं कर पाएंगे।

**problem of communication:-** यह बहुत ही बड़ा issue है इसमें miscommunication, language barriers, गलत assumption आदि आते हैं। अगर हम user को समझ ही नहीं पा रहे कि वो कहना क्या चाहता तो यह बहुत बड़ी बाधा बन जाता है।

## **Method (techniques) of requirement elicitation**

requirement elicitation के निम्नलिखित तरीके हैं -

**1:- Interview -** interview method के द्वारा यह समझा जाता है कि software से customer की क्या उम्मीदें हैं। इस method में end-user और stackholder से interview लिया जाता है यह interview software की कार्यसूची के base पर लिया जाता है और software से सम्बंधित सभी

requirements को collect करते हैं। इसमें पहले user से interview लिया जाता है उसके बाद stackholder से लिया जाता है।

**2:- Scenario :-** scenario एक step by step process होता है scenario का मतलब steps की series होती है। scenario में यह देखा जाता है कि किस प्रकार की और कितनी input की requirement होगी। इसमें user की जरूरत के अनुसार step by step information या requirement को collect किया जाता है।

**3:- Questionnaire :-** questionnaire जो है वह requirement collect करने का सबसे अच्छा method है। इसमें कम समय और कम cost में सॉफ्टवेयर से सम्बंधित महत्वपूर्ण information को collect करते हैं। इस method में एक बहुत बड़ी problem होती है कि सब अलग अलग information देते हैं क्योंकि सबका अलग अलग field होता है सब अपने अपने department का पूरा knowledge देते हैं। और सभी information को collect करके arrange किया जाता है इसमें सभी information सही है या जरूरी नहीं होता है। जो information सही होता है उसका use किया है।

**4:- On side observation: -** किसी company या organization जहाँ software को develop किया जा रहा है वहाँ जाकर information collect करना on side observation कहलाता है।

**5:- Prototype :-** किसी भी software को बनाने के लिए use में आने वाले component के बारे में हमें कोई जानकारी नहीं होती है। इसलिए उस component के बारे में पूरी information collect करने के लिए [prototype model](#) का use किया जाता है।

**निवेदन:-** आपको यह पोस्ट अच्छी लगी हो तो इसे अपने दोस्तों के साथ share करें. तथा requirement elicitation के बारे में कोई सवाल हो तो कमेंट में पूछ सकते हैं.

---

## requirement validation in hindi (software engineering)

by yugal joshi - Friday, August 17, 2018

<https://ehindistudy.com/2018/08/17/requirement-validation-hindi/>

### Requirement validation in hindi

software designing और development में किसी भी तरह का कोई problem ना आये इसलिए requirement validation का use किया जाता है। software को बनाने के लिए जो [requirements इक्कठा](#) की जाती है उन्हें analyze किया जाता है कि वह requirements हमारे software बनाने के लिए सही है या नहीं तथा software development के लिए कौन से process का प्रयोग किया जायेगा उसे भी analyze करते हैं।

Requirements और process को analyze करने के बाद एक [SRS \(Software requirement specification\)](#) बनाया जाता है। SRS बनाने के बाद यह सुनिश्चित करना जरूरी होता है कि SRS में कोई भी error नहीं है और यह specify किया जाता है कि user की requirements correct हैं।

SRS में ज्यादातर errors present होते हैं। यदि यह error development process के बाद या फिर user को software deliver करने के बाद detect हुए तो software के cost पर असर पड़ता है। इसलिए SRS से software designing और development के पहले error detect कर लेना चाहिये। requirement से सम्बंधित जितनी भी परेशानी होती है उन सभी को check करने के लिए requirement validation परफॉर्म किया जाता है।

SRS में जो भी ambiguity, data redundancy, error / bug की problem है उसे software development के पहले solve कर लेना चाहिये। इससे हमारे requirements जो हैं वह error free रहते हैं और software भी अच्छी quality का बनता है।

## Inputs for Requirement validation in hindi:-

यहाँ requirement validation के लिए तीन inputs होते हैं -

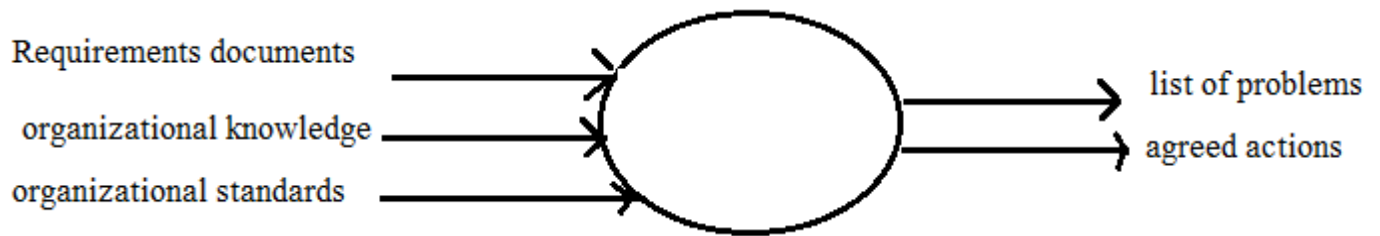
1. Requirement documents
2. Organizational knowledge
3. Organizational standard

**Requirement documents:-** organization के standard के अनुसार requirement document को formulated और organized होना चाहिये। हर organization का अपना अलग अलग standard होता है इसलिए organization के standard के अनुसार ही requirement document को बनाना चाहिये।

**Organizational knowledge:-** organizational knowledge का प्रयोग organization के बारे में पूरी जानकारी रखने के लिए, software development का process ठीक से चल रहा है या नहीं तथा system की requirements को जानने के लिए किया जाता है।



**Organizational standards :-** organizational standards में हमें organization के सभी rules, regulations तथा limits पता होनी चाहिये जिसे follow करके system develop कर सकें।



## Output of requirement validation in hindi:-

यहाँ requirement validation के दो output होते हैं -

1. List of problems
2. Agreed actions

**List of problem:-** list of problems में उन सभी problems की list होती है जिन्हें requirement document में discover किया था.

**Agreed action :-** agreed action एक list होती है। problem list में जितनी भी problems होती है उनको solve करने के लिए जो action परफॉर्म होती है उनको agreed action में display किया जाता है।

**Problem Aries during requirement validation (requirement validation में आने वाली परेशानियां) :-**

1. Data redundancy / repetition
2. Accuracy
3. Timeliness
4. Error / bug
5. Ambiguity

## Requirements Validation Tools & techniques in hindi:-

requirement validation की कुछ tools तथा techniques होती है इनका प्रयोग अकेले भी किया जा सकता है या फिर दूसरे techniques के साथ भी किया जा सकता है. इनका प्रयोग पूरे सॉफ्टवेयर को या फिर सॉफ्टवेयर के parts को check करने के लिए किया जाता है.

कुछ tools and techniques नीचे दिए हुए हैं -

1. Test CASE generation
2. Automated consistency analysis
3. Prototyping

**निवेदन:** अगर आपको यह पोस्ट पसंद आयी हो तो अपने दोस्तों के साथ share करें तथा requirement validation से संबंधित कोई सवाल हो तो कमेंट के द्वारा पूछ सकते हैं.

---

## SAD MCQ in hindi multiple choice question

by yugal joshi - Friday, May 18, 2018

<https://ehindistudy.com/2018/05/18/sad-mcq-hindi/>

### MCQ of SAD (System Analysis design) in hindi (बहुविकल्पीय प्रश्न )

मैंने यहाँ पर कुछ महत्वपूर्ण SAD mcq दिए हैं जो कि प्रतियोगी परीक्षाओं में बहुत उपयोगी साबित हो सकते हैं तो चलिए पढ़ते हैं.

Q1. system study के अन्दर आता है:-

- existing system की स्टडी करना
- existing system का डॉक्यूमेंटेशन करना
- सिस्टम की कमियों को identify करना तथा नए लक्ष्य निर्धारित करना.
- उपरोक्त सभी

Ans:- (d)

Q2. निम्नलिखित में से स्ट्रक्चर डिजाइनिंग के लिए कौन सा प्राथमिक टूल का उपयोग किया जाता है.

- Structure chart
- [Data flow diagram](#)

- Program flow chart
- Module

Ans :- (A)

Q3. जो किसे समस्या को step by step (चरणबद्ध) हल करता है निम्न में से उसे किस नाम से जाना जाता है

- algorithm
- plan
- list
- इनमे से कोई भी नहीं

Ans:- (a)

Q4. [DDL](#) का पूरा नाम क्या है ?

- data description language
- data definition language
- database description language
- इनमे से कोई भी नहीं

Ans. (b)

Q5 निम्न में से प्रॉब्लम analysis करने के लिए किस चरण (phase) का उपयोग किया जाता है

- system design phase
- system analysis phase
- before system test
- उपरोक्त कोई भी नहीं

Ans.(b)

Q6. इनफार्मेशन फ्लो चार्ट क्या प्रस्तुत करती है

- get access of picture of the system
- decision path को show करना.
- सिस्टम को graphically प्रस्तुत करना.
- इनमे से कोई भी नहीं

Ans :- (c)

Q7. निम्नलिखित में से hierarechical डाटा स्ट्रक्चर का उदाहरण कौन सा है

- ट्री
- लिंक लिस्ट
- array
- इनमें से कोई भी नहीं

Ans:- (a)

Q8. HIPO का पूरा नाम क्या है

- hierarchy input process output
- hierarchy input plus output
- hierarchyplus input process output
- इनमें से कोई भी नहीं

Ans:- (a)

Q9 flow chart के लाभ क्या है

- effective communication
- effective analysis
- represent the relations
- उपरोक्त सभी

Ans:- (d)

Q10 निम्न में से DDS का पूरा नाम hai

- data digital system
- data detail system
- data dictionary system
- digital data system

Ans:- (c)

SAD mcq

Q11. DFD में rectengle symbol का उपयोग किस लिए किया जाता है

- process
- data store

- external entity
- कोई भी नहीं

Ans:- (b)

Q12 [SDLC](#) का पूरा नाम क्या है

- system development life cycle
- structure design life cycle
- system database life cycle
- इनमें से कोई भी नहीं

Ans:- (a)

Q13. निम्न में से acceptance testing कब किया जाता है,

- running the system with line data by the actual user.
- testing change made in an existing or a new program.
- Is checking the logic of one or making program in the candidate system
- उपरोक्त सभी में

Ans:-(d)

Q14. निम्न में से documentation कब तैयार (prepared) किया जाता है

- सभी stages में
- system designing में
- system analysis में
- system development में

Ans:- (a)

Q15. SDLC (सिस्टम डेवलपमेंट लाइफ साइकिल) में कितने steps होते हैं?

1. 4
2. 6
3. 3
4. 5

ans: 6 स्टेप्स होते हैं.

Q16. निम्न में से system analysis के लिए किस का उपयोग नहीं किया जाता है ,

- flow chart
- decision टेबल
- system test डाटा
- dfd

Ans:- (c)

Q17. निम्न में से system डिजाइनिंग चरण में कौन से टूल का उपयोग नहीं किया जाता है,

- pie chart
- decision table
- system flow chart
- डाटा flow diagram

Ans:- (a)

Q18. किसी भी input record या किसी system पर अपने original स्रोत पर वापस की गयी प्रक्रिया का पता लगाना क्या कहलाता है ,

- audit trial
- conversion
- batch प्रोसेसिंग
- रिपोर्ट generation

Ans:- (a)

Q19. decision table का उपयोग क्यों करते हैं ,

- इनफार्मेशनल डेटा को tabular form में प्रस्तुत करने के लिए
- decision path को प्रदर्शित करने के लिए
- सिस्टम के सही चित्र को प्राप्त करने के लिए
- उपरोक्त कोई भी नहीं

Ans :- (a)

Q20 database system का लाभ क्या है,

- डेटा सभी एप्लीकेशन share कर सकते हैं.
- डेटा redundancy कम हो जाती है.
- डेटाबेस में उपस्थित डेटा को हम आसानी से एक्सेस कर सकते हैं
- उपरोक्त सभी

Ans:- (d)

SAD mcq

Q21. निम्न में से कौन सा डेटा फाइल, एक से ज्यादा (multiple data file) को contain करता है,

- इंडेक्स फाइल
- sequential file
- inverted file
- index - sequential file

Ans:- (d)

Q22. निम्नलिखित file में से least file कौन सा है,

- work file
- master file
- program file
- transaction file

Ans:- (a)

Q23. SDLC का प्रथम स्टेप कौन सा है?

1. analysis
2. design
3. documentation
4. problem identification

ans:- problem identification

Q24. cost benefit analysis क्या है

- हार्डवेयर तथा सॉफ्टवेयर की cost को estimate करना.
- tangible तथा intangible घटकों का मूल्यांकन करना.
- cost को benefit के साथ compare करना.

- उपरोक्त सभी

Ans:- (d)

Q25. निम्न में से सिस्टम के characteristics कौन सा है,

- intraction
- organization
- interdependence
- उपरोक्त सभी

Ans:- (d)

Q26. निम्न में से सिस्टम का type कौन सा नहीं है ,

- open and close
- physical and abstract
- man-made informational system
- उपरोक्त सभी

Ans:- (d)

Q27. निम्न में से कौन सा इनफार्मेशन सिस्टम को ग्राफिकली प्रस्तुत करता है,

- histogram
- pictogram
- flow chart
- DFD( data flow diagram)

Ans:- (d)

Q28. audit trial का उपयोग क्यों करते हैं,

- detail tracing के लिए
- सिस्टम में डेटा कैसे change होता है यह देखने के लिए.
- both (a) &(b)
- store डाटा

Ans:- (c)

Q29. benchmark क्या है,



- table technique
- discover करने का तरीका है
- compared two compititor
- उपरोक्त सभी

Ans:- (d)

Q30. [coupling](#) क्या होती है,

- relations
- दो modules के मध्य relation
- module के elements के मध्य relation
- उपरोक्त कोई भी नहीं

Ans:- (b)

SAD mcq

Q31. coheshing क्या है

- coupling का भाग
- module का एक भाग
- module के elements के मध्य relation
- उपरोक्त सभी

Ans:- (c)

Q32. डिजाईन तथा डेटाबेस स्ट्रक्चर को implement कौन करता है?

1. प्रोग्रामर्स
2. टेक्निकल writers
3. प्रोजेक्ट managers
4. डेटाबेस एडमिनिस्ट्रेटर

ans:- डेटाबेस एडमिनिस्ट्रेटर

Q33:- Debugging है:-

1. प्रोग्रामिंग करना
2. रिक्वायरमेंट्स को इक्कठा करना
3. अल्गोरिथम बनाना
4. errors को ढूँढना तथा उन्हें सही करना.

ans:- errors को ढूँढना तथा उन्हें सही करना.

Q34:- [DSS \(decision support system\)](#) किसके द्वारा प्रयोग किया जाता है?

1. line managers
2. top level managers
3. middle level managers

ans:- top level managers

निवेदन:- अगर आपको SAD MCQ पसंद आये हों तो कमेंट के द्वारा बताइए तथा इसे अपने दोस्तों के साथ share करें.

---

## software engineering testing question paper Msc 2017

by yugal joshi - Saturday, January 13, 2018

<https://ehindistudy.com/2018/01/13/software-engineering-testing-question/>

### software engineering testing exam question paper MSc madhya pradesh

software engineering testing exam question

#### UNIT I

प्रश्न 1. (A):- Project matrices का मतलब क्या है सॉफ्टवेर मापने के लिए उपयोग किये गये दो matrices का वर्णन कीजिये ?

(B):- [सॉफ्टवेर की विशेषताओं](#) और उपयोगों का वर्णन करें ?

अथवा

(C):- सॉफ्टवेर माप (software measurement) परिभाषित कीजिये ? सॉफ्टवेर इंजीनियरिंग में एक लेयर तकनीक को समझाये?

(D):- [सॉफ्टवेर इंजीनियरिंग](#) का क्या अभिप्राय है ? वर्तमान समय में इसकी आवश्यकता समझाइये?

## UNIT II

प्रश्न 2. (A):- Project planning से क्या अभिप्राय है? project planning के लिए निर्धारित कार्य क्या है ?

(B):- [software requirement specification](#) (सॉफ्टवेर की आवश्यक विशिष्टता) है? सॉफ्टवेर आवश्यकता विशिष्टता की विभिन्न विशेषताओं को समझाइये?

अथवा

(C): - उधारण के साथ process base estimation ( प्रक्रिया-आधार आकलन) समझाइये?

(D):- project scop से आपका क्या अभिप्राय है ? scop क लिए कौन –कौन से जानकारी की आवश्यकता है ?

## UNIT III

प्रश्न 3. (A):- सॉफ्टवेर डिजाईन में विभिन्न डिजाईन सिद्धांतों को विस्तार से समझाओ ?

(B):- अगर किसी मौजूदा मोडूल्स को एक नई प्रणाली बनाने में फिर से इस्तेमाल किये जाना है,तो किस डिजाईन रणनीति का इस्तेमाल किया जाता है और क्यों?

अथवा

(C): - effective modular design (प्रभावी मॉड्यूलर डिजाइन) से क्या अभिप्राय है ?  
चर्चा कीजिये?

(D):- निम्नलिखित को समझाये

1. internal or external design (आंतरिक or बाहरी डिजाइन)
2. procedural design (प्रक्रियात्मक डिजाइन )

#### UNIT IV

प्रश्न 4. (A) [unit testing](#) (इकाई परीक्षण) से आपका क्या अभिप्राय है?

b) [software testing](#) (सॉफ्टवेयर परीक्षण ) क्या है ? [black or white box testing](#) (काला or सफेद बॉक्स परीक्षण) समझाये?

अथवा

a) Interface डिजाइन के लिए दिशा –निर्देश नियमों का वर्णन कीजिये?

b) validation testing (सत्यापन परीक्षण) से क्या अभिप्राय है ?

#### unit v

प्रश्न 5. (a) Case environment के सामान्य आर्किटेक्चर को ड्राफ्ट कीजिये और इसके लक्षणों की व्याख्या कीजिये?

(B) [Upper or lower case](#) tools क्या है, upper case tool का उद्देश्य क्या है  
समझाईये?

or

(C) Case environment के सामान्य आर्किटेक्चर को draw कीजिये और इसके characteristics की व्याख्य कीजिये?

(D) निम्नलिखित को समझाये

1. Case repository
2. Integrating architecture

**निवेदन:-** दोस्तों, अगर software engineering testing exam question का यह पेपर आपके लिये helpful रहा हो तो हमें comment के द्वारा बताइए तथा इसे अपने दोस्तों के साथ share करें.

---

## Spiral model in hindi

by yugal - Tuesday, December 15, 2015

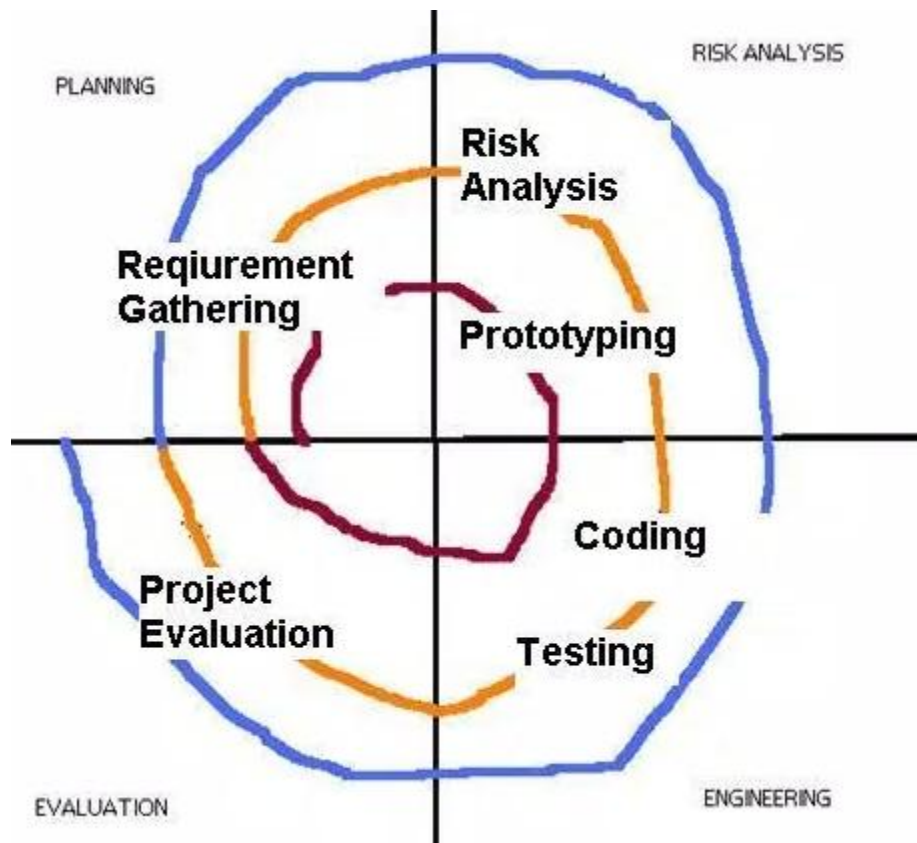
<https://ehindistudy.com/2015/12/15/spiral-model-in-hindi/>

### Spiral model in hindi (स्पाइरल मॉडल):-

Spiral model को 1985 में BOHEM ने प्रस्तावित किया था। इस मॉडल का आकार घुमावदार(spiral) होने की वजह से इसे spiral model कहते हैं।

स्पाइरल मॉडल में [वॉटरफॉल मॉडल](#) तथा [प्रोटोटाइप मॉडल](#) दोनों आते हैं अर्थात यह वॉटरफॉल मॉडल तथा प्रोटोटाइप मॉडल दोनों का संयोजन है।

स्पाइरल मॉडल का प्रयोग बड़े projects के लिए किया जाता है, छोटे projects में इसका प्रयोग नहीं किया जाता है तथा यह मॉडल बहुत अधिक खर्चीला(expensive) है।



### spiral model phases in hindi:-

स्पाइरल मॉडल में निम्नलिखित 4 phases होते हैं:-

- 1:-Planning
- 2:-Risk analysis
- 3:-Engineering
- 4:-Evaluation

**1:-Planning:-** प्लानिंग फेज में जितनी भी हमारी requirements हैं उनको एकत्रित किया जाता है। planning phase में हम सॉफ्टवेयर को हम क्या achieve कराना चाहते हैं या उसके goals क्या हैं?, discuss करते हैं।

**2:-Risk analysis:-** इस फेज में हम जितने भी रिस्क हैं उनको identify किया जाता है तथा अगर कोई रिस्क मिलता है तो उसका solution निकाला जाता है।

**3:-Engineering:-** इस फेज coding तथा टेस्टिंग की जाती है तथा सॉफ्टवेयर के डेवलपमेंट की पूरी प्रक्रिया इसी फेज में आती है।

**4:-Evaluation:-** इस फेज में जो भी सॉफ्टवेयर बनके तैयार हुआ है उसको customer evaluate(मूल्यांकन) करते हैं तथा अपना feedback देते हैं।

**निवेदन:-** आपको यह पोस्ट कैसी लगी हमें comment के द्वारा बताइए तथा इस पोस्ट को अपने दोस्तों के साथ share करें. धन्यवाद.

---

## stress testing in hindi

by yugal joshi - Saturday, December 16, 2017

<https://ehindistudy.com/2017/12/16/stress-testing-hindi/>

## Stress testing in hindi:- (स्ट्रेस टेस्टिंग)

stress testing का प्रयोग सिस्टम की stability (स्थिरता) तथा reliability (विश्वसनीयता) को टेस्ट करने के लिए किया जाता है.

यह एक non functional टेस्टिंग तकनीक है जो कि performance testing का एक हिस्सा है.

stress testing में सिस्टम को overload करके उसको verify तथा check किया जाता है कि सिस्टम stress को झेल पा रहा है या नहीं.

यह टेस्ट मुख्यतया यह निर्धारित करता है कि सिस्टम कितना robust (शक्तिशाली) है तथा heavy load में भी सिस्टम errors को कैसे हैंडल करता है.

stress testing यह भी सुनिश्चित करने के लिए किया जाता है कि संकट के समय में कहीं सिस्टम crash तो नहीं कर जाएगा.

स्ट्रेस टेस्टिंग को endurance testing तथा torture testing भी कहते हैं.

इसे भी पढ़ें:- [security testing](#) किसे कहते हैं?

[recovery testing](#) किसे कहते हैं?

## **stress testing क्यों की जाती है?**

इस टेस्टिंग को करने के निम्नलिखित कारण हैं:-

- सिस्टम की परफॉरमेंस को check करने के लिए.
- यह verify करने के लिए कि जब सिस्टम crash होता है तो वह डेटा को save करता है या नहीं.
- यह check करने के लिए जब सिस्टम crash होता है तो वह error message दिखाता है या नहीं.
- यह check करने के लिए जब सिस्टम failure होता है तो security बनी रहती है या नहीं.

## **stress testing कैसे की जाती है?**

- एक ही समय पर बहुत सारे users सिस्टम को run करते हैं.
- सभी users एक ही समय में सिस्टम पर कठिन operations करते हैं.
- सभी users एक ही समय में एक फाइल को खोलते हैं.

निवेदन:- अगर आपको पोस्ट अच्छी लगी हो तो हमें कमेंट के द्वारा बताइए तथा इसे अपने दोस्तों के साथ share करें.

---

## **system modeling in hindi & UML, types of system models in hindi**

by yugal joshi - Monday, August 06, 2018



## System modeling in hindi

system modeling को system model भी कहते हैं.

system modeling एक process है। यह प्रोसेस एक system के संपूर्ण structure को develop करता है। यह प्रोसेस system के abstract model को develop करता है, इसमें प्रत्येक model सिस्टम के एक अलग-अलग view या perspective को प्रस्तुत करते हैं।

system modeling में बहुत से models सम्मिलित होते हैं अर्थात् यह कई सारे model से मिलकर बना होता है। system model में जितने भी models होते हैं उन्हें graphical way (चित्रों के रूप) में represent किया जाता है।

हम किसी भी system के संपूर्ण model या structure को represent करने के लिए graphical structure या graphical Notation का प्रयोग करते हैं जो एक language पर आधारित होता है इस language को UML (unified modeling language) कहते हैं।

system modeling जो है वह सिस्टम तथा मॉडल की functionality को analysis करके समझने में मदद करता है। जो कि customer के साथ communicate करने के लिए प्रयोग किया जाता है।

## System perspective –

system modeling में system perspective के चार प्रकार होते हैं -

1. External perspective
2. Interaction perspective
3. Structural perspective
4. Behavioral perspective

**External perspective** - external perspective वो perspective है जिसका प्रयोग system के एक external environment को model या design करने के लिए किया जाता है।

**Interaction perspective** - एक interaction perspective वो perspective है , जहाँ हम एक system और उसके environment के बीच या फिर एक system के components के बीच interaction को model (डिजाइन) करते हैं।

**Structural perspective** - एक structural perspective वह perspective है , जहां आप सिस्टम के organization या सिस्टम द्वारा processed डेटा के structure का model करते हैं।

**Behavioral perspective** - behavioral perspective वो perspective है , जहाँ हम सिस्टम के dynamic behaviour को design करते हैं। और यह events को कैसे respond करता है यह देखते हैं।

## UML in hindi

UML का पूरा नाम unified modeling language है। जो एक object Oriented design methodology में real world objects को represent करने के लिए एक standard है। अर्थात् UML एक स्टैंडर्ड modeling language है जो कि diagrams के समूह को contain किये रहता है. जिससे developers को model बनाने में मदद मिलती है.

## UML digram types -

यहाँ पाँच प्रकार का UML digram है जो किसी भी organization के लिए system design या Architecture को represent करता है।

1. **Activity digram** - activity digram किसी process में या data processing में शामिल activities को show करता है।
2. **Use-case digram** - Use-case digram एक system और उसके environment के बीच interaction को show करता है।
3. **Sequence digram** - sequence digram जो है जो है वह system तथा system के components के बीच interaction को show करता है।
4. **Class digram** - class digram सिस्टम में object class और इन classes के बीच association को show करता है।
5. **State digram** - state digram यह show करता है कि system internal और external events पर कैसे react करता है.

## Types of system models in hindi

system models के कई प्रकार हैं जिसमें कुछ models नीचे describe किये गए हैं -

1. Data model
2. Object model

3. Context model
4. Behavioral model

## 1:- Data model -

data model में हम यह represent करते हैं कि हमारा data कैसे एक module से दूसरे module में move कर रहा है। data modeling , system model का एक प्रकार है। data model में input , output और processing data के flow को graphical Notation के द्वारा represent किया जाता है। इसे E-R modeling या E-R Notation भी कहते हैं।

### Data के flow के लिए Notation का भिन्न - भिन्न प्रकार (data flow notation) -

- **Rectangle shape** - इस shape का use किसी भी system के external entities को represent करने के लिए करते हैं।
- **Data store** - इस shape का use data base में कई प्रकार के information या data को store करने के लिए किया जाता है।
- **Data process / circle** - इस shape का use इस Notation में किसी भी information या data के processing के लिए किया जाता है।
- **Diamond shape / decision making digram** - इस shape का use entity के बीच relationship को represent करने के लिए किया जाता है।
- **Arrow** - इस shape का प्रयोग left to right , right to left , top to bottom or bottom to top डेटा के movement को represent करने के लिए किया जाता है।

### Perception (SCHEMA) of data model -

data model को प्रस्तुत करने के लिए यहाँ तीन प्रकार का perception है -

- **Physical perception** - इसमें [database schema](#) यह create और check करता है कि हमारा database schema एक दुसरे से कैसे related है।
- **Conceptual perception** - इसे logical perception भी कहते हैं। इसमें यह देखते हैं कि हमारा data कैसे insert , delete और update हो रहा है।
- **External perception** - इसे end user perception भी कहते हैं। इसमें यह देखते हैं कि हमारा data कैसे और किस environment में processing कर रहा है।

## 2:- Object model in hindi -

Object model एक system model का प्रकार है। जिसमें object का प्रयोग करके सभी real world problems को solve किया जाता है।

object model जो है वह object, classes तथा उनके मध्य association को system को describe करता है। object model में complex project को सरलता से पूरी Security के साथ विकसित कर सकते हैं।

object model में classes का प्रयोग data information को और उनके associated function को bind करने के लिए करते हैं। इस model में हम irrelevant या unnecessary information को end users से छुपा करके रखते हैं।

विभिन्न object model का प्रयोग object Oriented approach produce करने के लिए करते हैं जो निम्न है:-

- **Inheritance model** - इस model में एक class के information या attribute दूसरे class से inherit करते हैं। एक class जिससे information दूसरे class द्वारा inherit की जाती है उसे super class कहते हैं और वह class जो एक class से information को inherit करता है उसे sub class कहते हैं।
- **Aggregation model** - aggregation model में हम यह show करेंगे की कैसे classes दूसरे class से घिरी हुई रहती है। इस model में हम classes के बीच relationship को represent करेंगे।
- **Interaction model** - इस model में हम यह show करते हैं कि कैसे हमारा data एक object से दूसरे ऑब्जेक्ट में move करता है। interaction model में , data का एक object से दूसरे object पर movement करना interaction model की अहम भूमिका है।

### 3:- Context model -

context model एक graphical representation होता है। इसमें कुछ basic बातों को decide करना जरूरी होता है जैसे cost और time। इसमें यह भी decide किया जाता है कि सिस्टम में क्या क्या functionality डालनी है तथा इसमें यह बताया जाता है कि एक particular element को कैसे displayed करना है।

context model का प्रयोग एक सिस्टम के operational context को स्पष्ट करने के लिए किया जाता है। यह system उस environment को represent करता है जिसमें सिस्टम work कर रहा

होता है। यह system के requirement का use करके boundary decide करता है। इसका मुख्य लक्ष्य किसी भी system में context information के formal या semiformal description को produce करना होता है।

#### 4:- Behavioral model -

behavioral model सिस्टम मॉडल का एक प्रकार है। behavioral model का प्रयोग किसी भी system के संपूर्ण behavior को describe करने के लिए किया जाता है। इस model में हम एक state से दूसरे state तक data flow को describe करते हैं और प्रोसेस के execution के दौरान state के changes को भी describe करते हैं।

किसी भी system के behavior को describe करने के लिए निम्नलिखित models का use किया जाता है -

- **Data processing model** - data processing model सिस्टम में data के flow को describe करता है। हम data के flow को describe करने के लिए DFD का use करते हैं। [DFD का पूरा नाम data flow digram है।](#) DFD एक graphical Notation है जिसका use system में data के flow को represent करने के लिए किया जाता है।
- **State machine model** - state machine model किसी भी system के behavior को describe करता है। यह external और internal events के लिए response देता है। state machine model, सिस्टम state को node के रूप में और event को arcs के रूप में describe करते हैं। जब एक event occur होता है तब system एक state से दूसरे state में move करता है।

**निवेदन:-** अगर आपको system modeling की यह पोस्ट अच्छी लगी हो तो मुझे कमेंट के द्वारा बताइए तथा इसे अपने दोस्तों के साथ share करें. धन्यवाद.

---

## TQM क्या है तथा इसके लाभ क्या है?

by yugal joshi - Friday, July 15, 2016

<https://ehindistudy.com/2016/07/15/total-quality-managementtqm-hindi/>

यहाँ पर आप what is TQM in hindi (टोटल क्वालिटी मैनेजमेंट क्या है? पढ़ेंगे

# TOTAL QUALITY MANAGEMENT (TQM)

## in hindi

TQM एक ऐसी मैनेजमेंट तकनीक या मेथड है जिसमें कि processes की क्वालिटी, प्रोडक्ट की क्वालिटी तथा सभी क्रियाकलापों(activities) को बेहतर बनाने के लिए लगातार प्रयास किये जाते हैं.

आसान शब्दों में कहें तो "यह एक ऐसी प्रक्रिया है जिसमें मैनेजमेंट तथा संगठन/कंपनी के सभी कर्मचारियों के द्वारा customer को बेहतर सर्विस देने के लिए लगातार प्रयास किये जाते हैं."

organisation में उपर से लेकर नीचे तक के कर्मचारियों की भूमिका क्वालिटी प्रोडक्ट उपलब्ध कराने में तथा customer को अच्छी सर्विस देने में होती है तथा TQM किसी एक विशेष प्रोजेक्ट तथा प्रोडक्ट पर आधारित नहीं होता है बल्कि यह पूरी organisation पर आधारित होता है.

यह एक ऐसी प्रक्रिया है जिसमें errors तथा defects को हटा दिया जाता है तथा customer को बेहतर सुविधा उपलब्ध करायी जाती है.

कभी भी यह याद रखना चाहिए कि कोई भी customer जब किसी कंपनी के प्रोडक्ट से खुश या satisfy होता है तो वह उस प्रोडक्ट तथा कंपनी के बारे में और लोगों को बताता है तो ज्यादा लोग उससे जुड़ते हैं. इसलिए total quality management बहुत जरूरी है.

TQM सबसे पहले customer को ध्यान में रखकर शुरू होता है और कभी खत्म नहीं होता है क्योंकि इसके जीवनकाल का commitment ही क्वालिटी को लगातार बेहतर बनाना होता है.

इसे William Deming ने 1980 में विकसित किया था तथा इसे हिंदी में "सम्पूर्ण गुणवत्ता प्रबन्धन" कहते हैं.

- [customer relationship management \(CRM\) क्या है?](#)
- [supply chain management \(SCM\) क्या है?](#)

## TQM process in hindi

इसकी प्रक्रिया 4 भागों में विभाजित होती है:-

1:-PLAN 2:-DO 3:-CHECK 4:-ACT अर्थात PDCA cycle(चक्र).

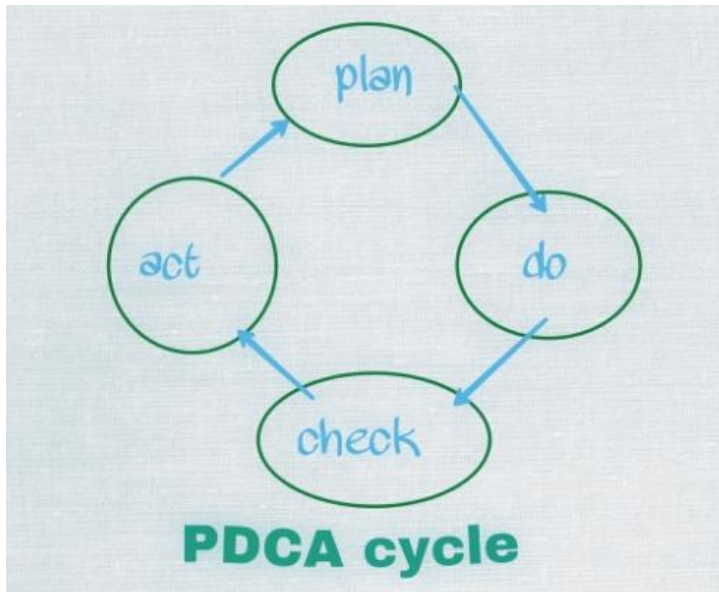


Fig:-pdca cycle

**1:-PLAN:-** यह PDCA चक्र का पहला स्टेप होता है इसमें परेशानियों का आकलन किया जाता है और इन परेशानियों के आधार पर नये plan बनाये जाते हैं.

**2:-DO:-** इस स्टेप में जो plan बनाया गया था उसको implement किया जाता है. और जो भी बदलाव किये जाते हैं उनको मूल्यांकित करने के लिए documented कर लिया जाता है.

**3:-CHECK:-** इस स्टेप में पिछले phase से प्राप्त डेटा की जांच की जाती है. इसमें यह देखा जाता है कि जो plan किया था उसको achieve कर लिया गया है या नहीं.

**4:-ACT:-** यह PDCA का आखरी स्टेप है इस स्टेप में organisation के अन्य कर्मचारियों के साथ communicate किया जाता है और उनसे discuss करके नये procedures को implement किया जाता है. तथा इस नये implementation की जानकारी अन्य लोगों को दी जाती है.

यहाँ याद रखने वाली बात यह है कि यह एक लगातार चलने वाला चक्र है act स्टेप के बाद फिर से plan स्टेप आ जाता है.

## TQM का प्रभाव

नीचे दिए गए चित्र के द्वारा हम TQM के प्रभाव को समझ सकते हैं:-

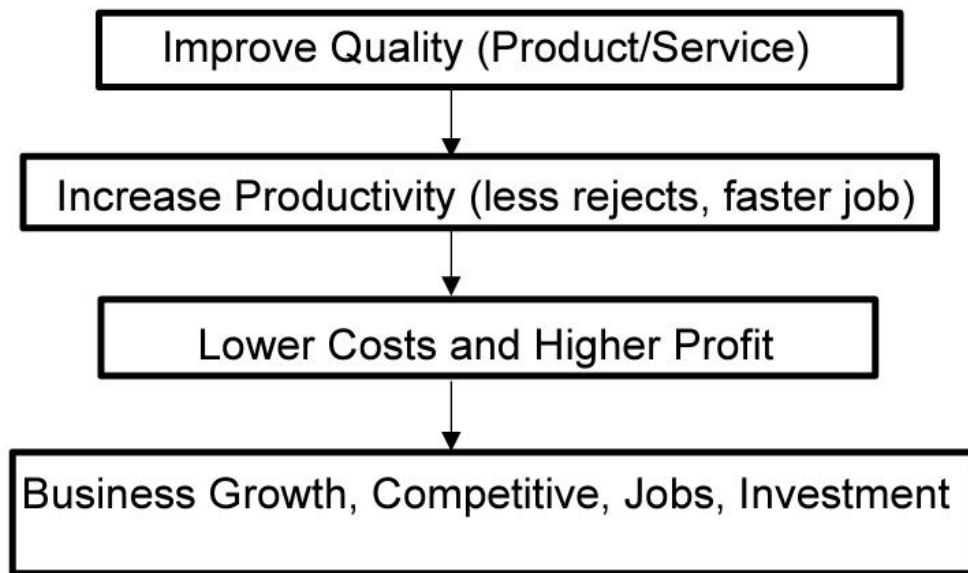


fig:-effect

of TQM

"अगर प्रोडक्ट की क्वालिटी बेहतर होगी तो productivity बढ़ेगी, productivity बढ़ेगी तो लागत में कमी आएगी और मुनाफा ज्यादा होगा जिससे business बढ़ेगा तथा jobs, investment एवम् competition बढ़ेगा।"

## PRINCIPLES of TQM in hindi

इसके सिद्धांत निम्नलिखित हैं:-

**1:-customer focus:-** कंपनी में customer की आवश्यकताओं के अनुसार तथा satisfaction के आधार पर निर्णय लिए जाते हैं और वैसे भी कोई भी customer तब satisfy होता है जब अच्छी क्वालिटी का प्रोडक्ट कम दामों में प्राप्त हो.

**“CUSTOMER IS KING”**

customer को राजा की तरह treat किया जाता है।

**2:- continuous improvement:-** TQM की एक और विशेषता सभी activities को और बेहतर करना है और यह लगातार जारी रहता है. प्रोडक्ट की क्वालिटी को बेहतर बनाने के लिए नियत तथा निरंतर प्रयास किये जाते हैं.

**3:- employee involvement:-** इसमें कर्मचारियों को शामिल किया जाता है क्योंकि कर्मचारियों के द्वारा ही प्रोडक्ट में क्वालिटी को बेहतर किया जा सकता है. क्योंकि अगर कर्मचारियों को



empower(सशक्त) बनाया जायेगा तो वे अपनी सर्विस बेहतर ढंग से दे पायेंगे.

“इसमें सभी लोगो को शामिल किया जाता है;’ manager से लेकर चौकीदार तक सभी लोगो को’ क्योंकि सभी लोगो के अपने roles तथा जिम्मेदारियां होती है.”

**4:- techniques तथा tools:-** कंपनी में तकनीक तथा tools का प्रयोग बहुत ही आवश्यक है; इनसे क्वालिटी को increase किया जा सकता है. ज्यादातर कर्मचारी एक समय में केवल एक ही टूल का प्रयोग करते हैं परन्तु कभी-कभी tools को combination में प्रयोग करना बहुत फायदेमंद साबित होता है.

**5:- decision-making:-** कंपनी में जो निर्णय लिए जाये वह तथ्यों तथा डेटा के आधार पर होने चाहिए न कि emotions तथा personal issues के आधार पर.

**6:- communication:-** किसी भी बिज़नेस के सफल होने के लिए कम्युनिकेशन बहुत आवश्यक है. अगर कस्टमरों तथा कर्मचारियों के मध्य ढंग से communication नहीं हो पा रहा है तो बिज़नेस का खाक होगा.

## advantage of TQM in hindi (टीक्यूएम के लाभ)

- इसका लाभ यह है कि इससे customer संतुष्ट रहता है क्योंकि कंपनी के पास प्रोडक्ट तथा सेवायें बेहतरीन होती हैं उनमें कोई गलती नहीं होती है. और तब कंपनी को बहुत मुनाफा होता है क्योंकि ग्राहक खुद उस प्रोडक्ट के बारे में अन्य लोगों को भी बताते हैं.
- इससे product की defects (कमियों) को कम किया जाता है. क्योंकि tqm का मुख्य कार्य है प्रोडक्ट की quality को बेहतर बनाना, जिससे प्रोडक्ट बहुत ही अच्छा बनता है.
- इससे यह पता चलता है कि market में किन product की जरूरत है.
- इससे कंपनी की cost में कमी आती है.

अगर यह पोस्ट tqm in hindi आपके लिए helpful रही है तो हमें comment के माध्यम से बताइये तथा इसे अपने दोस्तों के साथ share करें। धन्यवाद...

---

## Waterfall model in hindi

by yugal - Tuesday, December 08, 2015

## waterfall model in hindi:-

सॉफ्टवेयर इंजीनियरिंग के लिए waterfall model (वॉटरफॉल मॉडल) [SDLC\(सिस्टम डेवलपमेंट लाइफ साइकिल\)](#) का एक प्रसिद्ध और अच्छा version है। waterfall model (वॉटरफॉल मॉडल) linear तथा sequential मॉडल है इसका अर्थ यह है कि एक डेवलपमेंट phase तब तक शुरू नहीं हो सकता जब तक कि उसका पिछला वाला phase पूरा नहीं हो जाता है। हम वॉटरफॉल मॉडल में phases को overlap नहीं कर सकते हैं।

"हम वॉटरफॉल को निम्न तरीके से imagine कर सकते हैं:-

एक बार जब पानी चट्टान के किनारे के ऊपर से प्रवाहित होने लगता है और पहाड़ के नीचे की ओर गिरने लगता है और यह पानी ऊपर की ओर वापस नहीं जा सकता है।"

इसी प्रकार waterfall model भी कार्य करता है, एक बार डेवलपमेंट का एक phase पूरा हो जाता है तो हम अगले phase में चले जाते हैं लेकिन वापस पिछले phase में नहीं जा सकते हैं।

वॉटरफॉल मॉडल में, एक phase का आउटपुट दूसरे phase के लिए इनपुट की तरह कार्य करता है।

वॉटरफॉल मॉडल में निम्नलिखित 5 phases होते हैं (waterfall phases in hindi):-

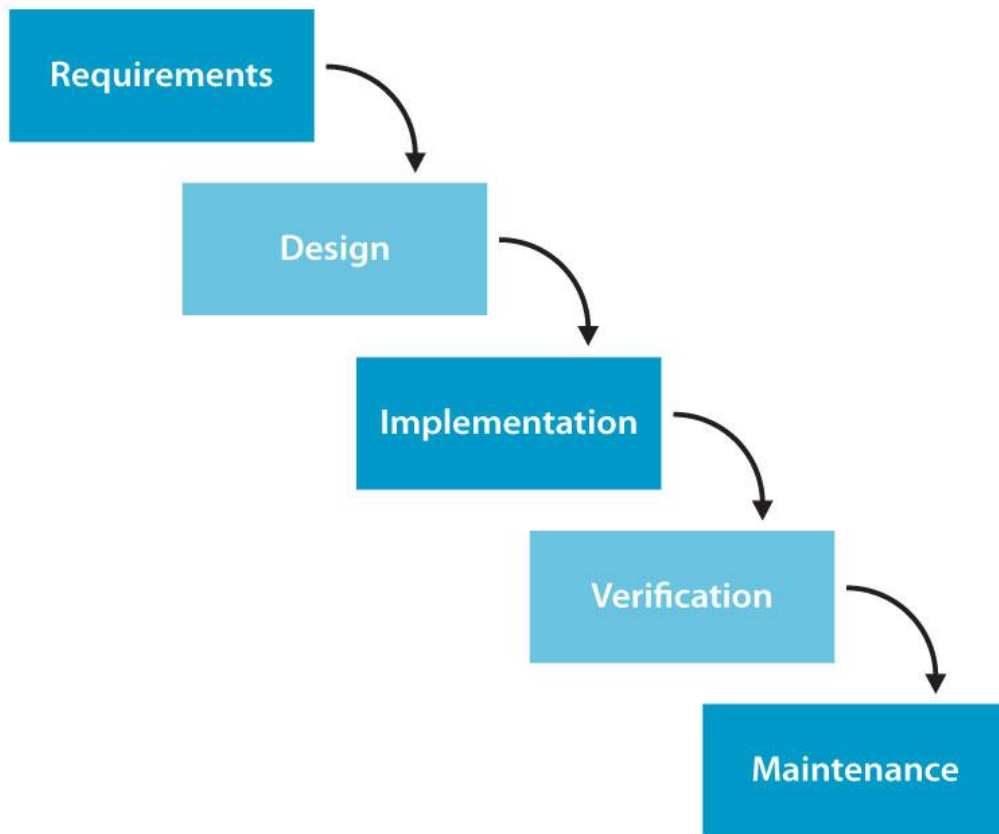


Fig:-waterfall model

**1:-Requirement phase:-** requirement phase वॉटरफॉल मॉडल का सबसे पहला phase है। इस फेज में सिस्टम की requirements को एकत्रित तथा documented किया जाता है। यह फेज बहुत crucial होता है क्योंकि इसी फेज पर अगले phase आधारित होते हैं।

**2:-Design phase:-** design phase इस तथ्य पर आधारित होता है कि सॉफ्टवेयर का निर्माण किस प्रकार होगा। डिज़ाइन फेज का मुख्य उद्देश्य सॉफ्टवेयर सिस्टम का blueprint तैयार करना है जिससे कि आने वाले phases पर किसी प्रकार की कोई दिक्कत का सामना ना करना पड़े और requirement phase में जो भी requirements हैं उनका solution निकाल लिया जाएँ।

**3:-Implementation phase:-** इस फेज में हार्डवेयर, सॉफ्टवेयर तथा एप्लीकेशन प्रोग्राम्स को install किया जाता है तथा डेटाबेस डिज़ाइन को implement किया जाता है। इससे पहले की डेटाबेस डिज़ाइन को implement किया जाए सॉफ्टवेयर को टेस्टिंग, कोडिंग, तथा debugging प्रोसेस से होकर गुजरना पड़ता है। वॉटरफॉल में यह सबसे लम्बे समय तक चलने वाला phase है।

**4:-verification phase:-** इस फेज में सॉफ्टवेयर को verify किया जाता है और यह evaluate किया जाता है कि हमने सही product बनाया है। इस फेज में विभिन्न प्रकार की टेस्टिंग की जाती है तथा सॉफ्टवेयर के हर area में check किया जाता है।

माना अगर हमने सॉफ्टवेयर को अच्छी तरह verify नहीं किया और इसमें कोई defect रह जाता है तो इसका इस्तेमाल कोई नहीं करेगा इसलिए verification अत्यंत महत्वपूर्ण है।

verification का एक advantage यह है कि इससे सॉफ्टवेयर के fail होने का risk कम हो जाती है।

**5:-Maintenance phase:-** यह वॉटरफॉल का सबसे अंतिम phase है। जब सिस्टम बनके तैयार हो जाता है तथा यूजर उसका प्रयोग करना शुरू कर देते हैं तब जो problems उसमें आती हैं उनको time-to-time हल करना पड़ता है। तैयार सॉफ्टवेयर को समय अनुसार उसका ख्याल रखना तथा उसे maintain रखना ही maintenance कहलाता है। SDLC में तीन प्रकार के maintenance होते हैं:-

1. corrective maintenance
2. adaptive maintenance
3. perfective maintenance.

इसे भी पढ़ें:- [स्पाइरल मॉडल](#)

निवेदन:- आपको यह waterfall model की पोस्ट कैसी लगी हमें comment के द्वारा बताइए तथा इसे अपने दोस्तों के साथ सहरे करें. धन्यवाद.

---

## what is basis path testing in hindi? सॉफ्टवेर टेस्टिंग

by yugal joshi - Monday, December 19, 2016

<https://ehindistudy.com/2016/12/19/basis-path-testing-hindi-%e0%a4%b8%e0%a5%89%e0%a4%ab%e0%a5%8d%e0%a4%9f%e0%a4%b5%e0%a5%87%e0%a4%b0-%e0%a4%9f%e0%a5%87%e0%a4%b8%e0%a5%8d%e0%a4%9f%e0%a4%bf%e0%a4%82%e0%a4%97/>

## White box testing techniques in hindi:-

### Basis Path Testing (बेसिस पाथ टेस्टिंग) in hindi:-

basis path testing एक [white box टेस्टिंग तकनीक](#) है जिसे 1980 में McCabe ने प्रस्तावित किया था.

इस टेस्टिंग का प्रयोग test cases को डिज़ाइन करने के लिए किया जाता है तथा इसमें test cases को डिज़ाइन करने के लिए प्रोग्राम के flow या logical path का प्रयोग किया जाता है.

यह टेस्टिंग सुनिश्चित करती है कि प्रोग्राम के सभी flow या logical path कम से कम एक बार execute कर लिए गये हैं. (इसके लिए यह McCabe के [cyclomatic complexity](#) का प्रयोग करता है.)

basis path testing में निम्नलिखित पद होते हैं:-

- 1:- सबसे पहले प्रोग्राम के विभिन्न path को निर्धारित करने के लिए control flow graph को बनाया जाता है.
- 2:- (independent paths को निर्धारित करने के लिए) cyclomatic complexity कैलकुलेट की जाती है.
- 3:- paths के एक basis समूह को ढूँढा जाता है.
- 4:- अंत में प्रत्येक path के लिए test cases को जनरेट किया जाता है.

basis path टेस्टिंग हमें प्रोग्राम के कोड में उपस्थित सभी त्रुटियों को निर्धारित करने में सहायक है.

---

## what is build and fix model in hindi? and its phases

by yugal joshi - Tuesday, July 31, 2018

<https://ehindistudy.com/2018/07/31/build-fix-model-hindi/>

## Build and fix model (ad hoc model in hindi):-

Build and fix model को ad hoc model भी कहते हैं। इसमें सॉफ्टवेयर को बिना किसी डिजाइन या specification के विकसित किया जाता है। और testing को भी इसमें ज्यादातर नजरअंदाज कर दिया जाता है।

यह सॉफ्टवेयर लाइफ साइकिल डेवलपमेंट का सबसे सरल model है। इसमें सॉफ्टवेयर बनाने के लिए कम requirements होती हैं।

इस model के द्वारा जब कोई शुरुवाती सॉफ्टवेयर बनके तैयार हो जाता है तो इसे user को deliver किया जाता है। user इसको check करता है अगर उसे सॉफ्टवेयर अच्छा नहीं लगता है तो सॉफ्टवेयर में functions को add, modify तथा delete किया जाता है और यह प्रक्रिया तब तक चलती है जब तक कि यूजर सॉफ्टवेयर से satisfy नहीं हो जाता है।

मगर सॉफ्टवेयर के डिजाइन में कमी तथा लगातार इसमें बदलाव करने की वजह से ऐसे सॉफ्टवेयर का user बहुत कम ही प्रयोग करते हैं। इस कारण सॉफ्टवेयर इंजिनियर इस development approach को बहुत कम use करते हैं।

## phases of build and fix model

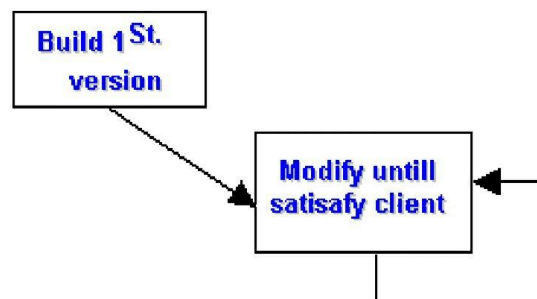


Figure: Build and Fix model

इस model में दो phases होते हैं:-

**1:- build:-** इस फेज में सॉफ्टवेयर code को विकसित किया जाता है और अगले फेज में भेज दिया जाता है.

**2:- fix:-** इस फेज में user की जरूरत के अनुसार code को correct तथा modify किया जाता है. तथा code को error free बनाया जाता है.

### **advantage of build and fix model (ad hoc model):**

इसके लाभ निम्नलिखित हैं:-

1:- इसमें सॉफ्टवेयर विकसित करने के लिए experience की बहुत कम जरूरत पड़ती है.

2:- इसमें सॉफ्टवेयर डेवलपमेंट के लिए गहरी planning नहीं करनी पड़ती.

3:- इसमें initial सॉफ्टवेयर तैयार हो जाता है.

### **disadvantage of build and fix model:-**

इसकी हानियाँ निम्नलिखित हैं:-

1:- इसमें सॉफ्टवेयर बनाने की cost बहुत अधिक होती है क्योंकि इसमें जब तक यूजर satisfy नहीं हो जाता है तब तक सॉफ्टवेयर में काम करना पड़ता है.

2:- सॉफ्टवेयर की क्वालिटी ना के बराबर होती है.

3:- इसमें risks बहुत होते हैं.

4:- इस सॉफ्टवेयर की maintenance करना बहुत मुश्किल है.

5:- इसमें सॉफ्टवेयर का डिजाईन informal होता है.

निवेदन:- अगर आपको यह पोस्ट अच्छी लगी हो तो कमेंट के द्वारा बताइए तथा इसे अपने दोस्तों के साथ share करें. धन्यवाद.

---

## What is capability maturity model(CMM) in hindi?

by yugal joshi - Wednesday, July 13, 2016

<https://ehindistudy.com/2016/07/13/capability-maturity-modelcmm-hindi/>

CMM का पूरा नाम capability maturity model है. यह एक ऐसा मॉडल है जिसका प्रयोग organisation(संगठन) के सॉफ्टवेर को विकसित तथा पहले से विकसित सॉफ्टवेर को improve(सुधारने) के लिए किया जाता है. इस model को सन् 1986 में SEI(software engineering institute) ने विकसित किया था.

बहुत से organisations में सॉफ्टवेर प्रोजेक्ट बहुत ही ज्यादा late हो जाते हैं और इनका बजट(budget) भी ज्यादा हो जाता है जिसके कारण प्रोजेक्ट undisciplined(अनुशासनहीन) तथा अव्यस्थित हो जाता है. इस परेशानी को हल करने के लिए SEI को CMM का निर्माण करना पड़ा था.

capability maturity model(CMM) जो है वह software organisation को मार्गदर्शित करती है कि सॉफ्टवेर को विकसित तथा maintain करने में जो processes इस्तेमाल में लायी जाती है उस पर नियन्त्रण कैसे किया जाएँ.

CMM सॉफ्टवेर प्रोजेक्ट के मूल्यांकन से प्राप्त knowledge तथा government एवम् industry से प्राप्त फीडबैक पर आधारित होता है. इसमें auditors के द्वारा organisations को उसकी क्षमता के अनुसार CMM रेटिंग दी जाती है.

आजकल इसको CMMI(capability maturity model intergation) कहा जाता है.

software process maturity के 5 levels होते हैं, जो निम्नलिखित हैं.



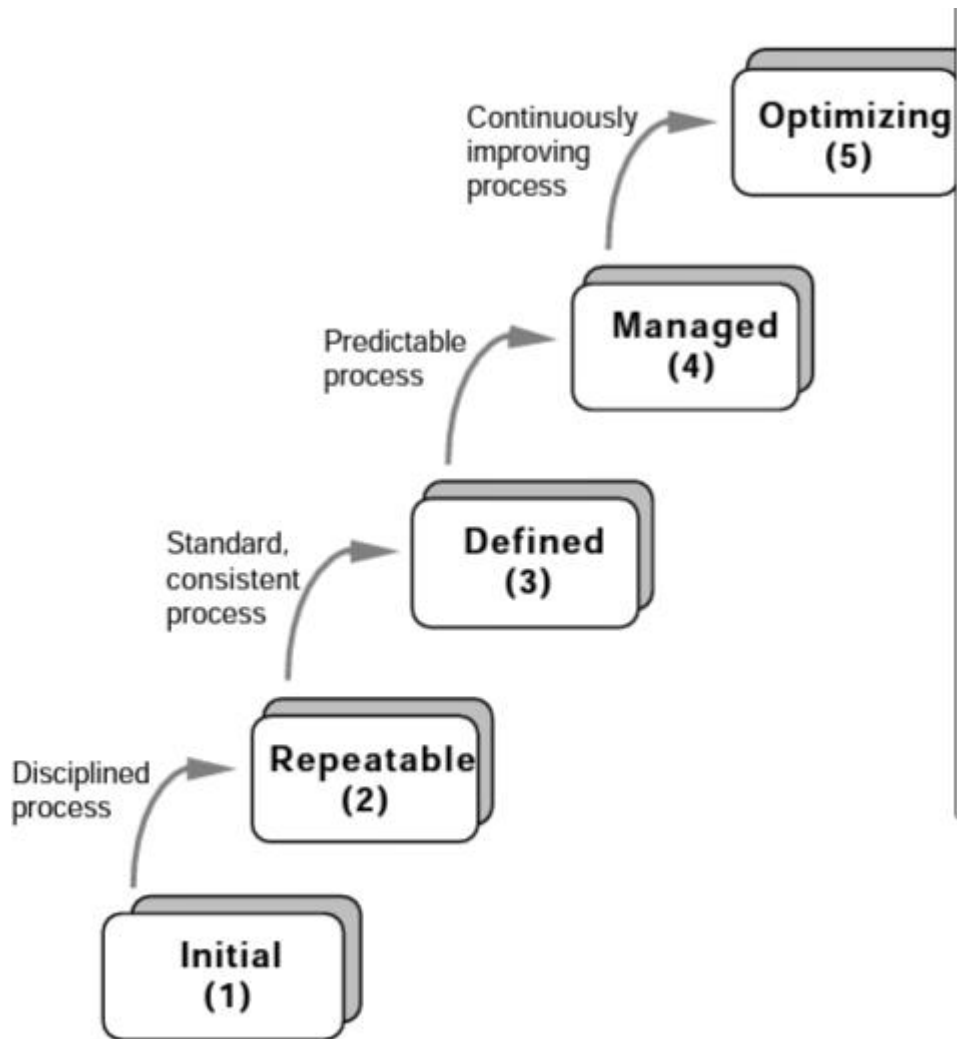


Fig:-levels of CMM

## 1:-initial level:-

यह cmm का पहला level है। यह लेवल बहुत ही unpredictable तथा बुरी तरह से controlled(नियंत्रित) होता है। इस level में सॉफ्टवेर को विकसित करने के लिए परिस्थितियां stable(स्थिर) नहीं होती हैं क्योंकि organisation का कोई standard नहीं होता है।

कोई भी सॉफ्टवेर प्रोजेक्ट तभी सफल हो सकता है जब manager तथा programmer अच्छी तरह काम करें और सफलता तब तक बनी रहेगी जब तक कि वह अगले प्रोजेक्ट में भी एक साथ काम करें। लेकिन इन दोनों में से कोई भी organisation को छोड़ देता है तो प्रोजेक्ट की क्वालिटी में गिरावट आ जाती है। प्रोजेक्ट की क्वालिटी में गिरावट का एक प्रमुख कारण सही तरीके से टेस्टिंग नहीं करना भी है।

## 2:-Repeatable level:-

यह cmm का दूसरा लेवल है. इसमें सॉफ्टवेर डेवलपमेंट कि सफलता के लिए process को दोहराया(reapeat) जाता है जो कि consistent परिणाम देते है.  
इसमें प्रोजेक्ट की लगातार सफलता के लिए प्रोजेक्ट मैनेजमेंट तकनीक का इस्तेमाल किया जाता है.  
इसमें नये प्रोजेक्ट की मैनेजमेंट तथा प्लानिंग समान प्रकार के प्रोजेक्ट्स के experience के आधार पर की जाती है.

### **3:-Defined level:-**

यह cmm का तीसरा level है. इसमें organisation के सॉफ्टवेर को विकसित तथा maintain करने में प्रयुक्त standard processes को documented तथा established(स्थापित) किया जाता है.  
इन standard processes को वक्त के साथ बेहतर बनाया जाता है जिससे कि सॉफ्टवेर प्रोजेक्ट की performance हमेशा consistence रहती है.  
processes के established होने से प्रोग्रामर, manager तथा अन्य technical staff अच्छी तरीके से कार्य कर सकते है तथा सॉफ्टवेर प्रोजेक्ट कि cost, schedule एवम functions को आसानी से नियंत्रित भी कर सकते है.

### **4:-Managed level:-**

यह CMM का चौथा लेवल है. इसमें organisation सॉफ्टवेर प्रोडक्ट तथा processes दोनों के लिए लक्ष्य निर्धारित करती है. इसमें प्रोजेक्ट कि क्वालिटी उच्च कोटि की होती है तथा performance हमेशा consistent तथा predictable होती है.  
इसमें सॉफ्टवेर डेवलपमेंट के लिए sub-process को select(चुना) किया जाता है जिनका कि पूरी processes की परफॉरमेंस में पूरा सहयोग रहता है.  
चुने गये sub-process को statistical तथा अन्य quantitative(मात्रात्मक) तकनीकों के द्वारा नियंत्रित किया जाता है.

### **5:-Optimizing level:-**

यह cmm का अंतिम यानि कि पाचवां लेवल है तथा इसमें process की improvment पर ध्यान दिया जाता है. process को बेहतर बनाने के लिए सम्भावित त्रुटियों तथा गलतियों को कम करता है तथा सॉफ्टवेर के विकसित होने में लगने वाली cost को भी कम करता है.

### **advantage of CMM:-**

CMM के कुछ लाभ निम्नलिखित हैं:-

- 1:-इसका एक लाभ यह है कि programmer, manager तथा technical staff के लोग हमेशा क्वालिटी को प्राथमिकता देते हैं.
- 2:-इसमें cost बहुत ही कम लगती है क्योंकि सॉफ्टवेर प्रोजेक्ट में defects तथा errors बहुत ही कम होती है.
- 3:-इससे सॉफ्टवेर प्रोडक्ट की productivity बढ़ती है.
- 4:-ज्यादातर customer(ग्राहकों) को लाभ होता है जिससे वह satisfy होते हैं.
- 5:-इसके द्वारा सॉफ्टवेर प्रोजेक्ट की predictability तथा consistency बढ़ती है.
- 6:-इसके द्वारा सॉफ्टवेर process में सुधार होता है.

अगर आपके लिए यह पोस्ट helpful रही है तो आप हमें comment के माध्यम से बताये तथा अपने दोस्तों के साथ पोस्ट को share करें. धन्यवाद.

---

## what is case tools in hindi?

by yugal joshi - Tuesday, November 14, 2017

<https://ehindistudy.com/2017/11/14/case-tools-hindi/>

## software Case tools in hindi:-

आज हम case tools के बारे में पढ़ेंगे, सबसे पहले हम समझेंगे कि case क्या होता है? तो चलिए शुरू करते हैं.

### CASE in hindi:-

case का पूरा नाम computer aided software engineering है, यह एक सॉफ्टवेयर है जो कि सॉफ्टवेयर को बनाने की प्रक्रिया में एक या एक से अधिक [सॉफ्टवेयर इंजीनियरिंग](#) गतिविधियों को सपोर्ट करता है.

सरल शब्दों में कहें तो, “case का मतलब है सॉफ्टवेयर को बनाने तथा maintain करने के लिए स्वचालित सॉफ्टवेयर tools का प्रयोग किया जाएगा.”

सॉफ्टवेयर को विकसित करने के लिए CASE बहुत ही प्रयोग किया जाता है क्योंकि इसका जो functionality तथा पोटेंशियल है वह बहुत ही अच्छा है जिससे [सॉफ्टवेयर की क्वालिटी \(गुणवत्ता\)](#) बढ़ती है.

## **CASE tools in hindi:-**

case tools सॉफ्टवेयर एप्लीकेशन का समूह है जो कि [सॉफ्टवेयर डेवलपमेंट लाइफ साइकिल \(SDLC\)](#) की गतिविधियों को automate (स्वचालित) कर देता है.

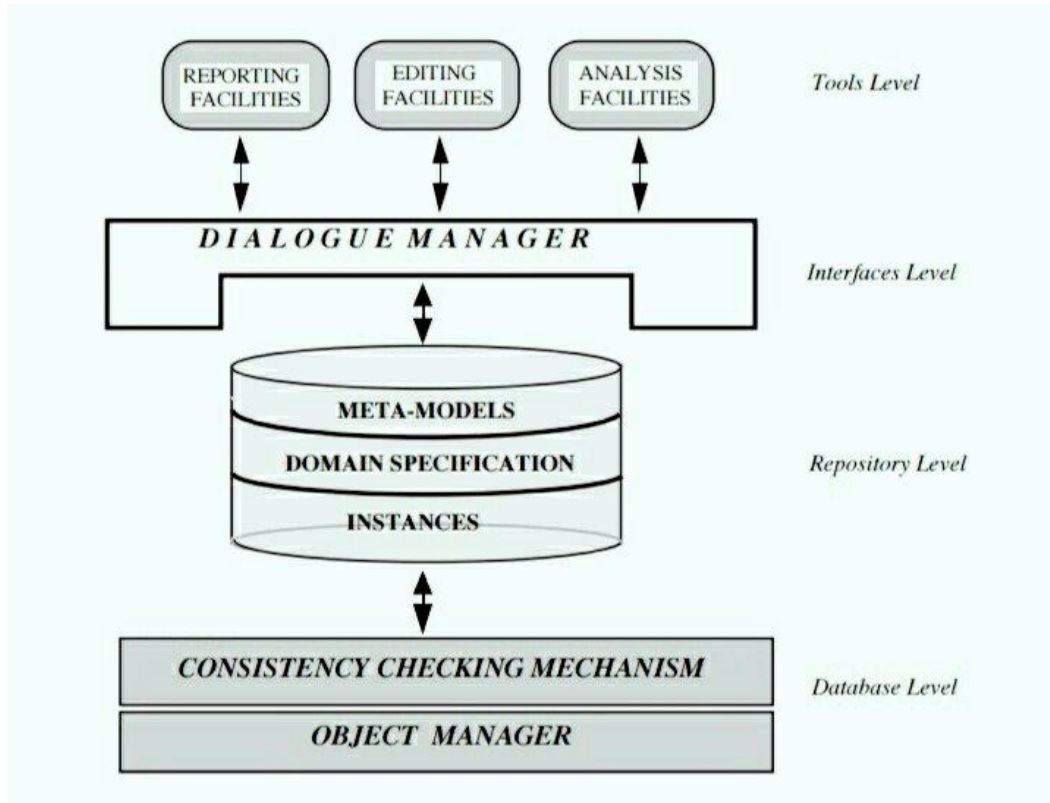
case tools का प्रयोग सॉफ्टवेयर इंजीनियर, सॉफ्टवेयर मैनेजर, तथा टेस्टर के द्वारा सॉफ्टवेयर को विकसित करने के लिए किया जाता है.

SDLC के ज्यादातर सभी phases जैसे:- डिजाईन, [टेस्टिंग](#), कोडिंग आदि को case tools सपोर्ट करते हैं. अर्थात् इन सभी phases में case tools प्रयोग किये जाते हैं. जो कि ये सुनिश्चित करते हैं कि सॉफ्टवेयर की क्वालिटी उच्च हों, उसमें कोई डिफेक्ट ना हों, तथा सॉफ्टवेयर डेवलपमेंट का काम किसी भी तरह कम मूल्य और समय में हो जाएँ.

case tools का प्रयोग debugging तथा डेटा को मैनेज करने के लिए भी किया जाता है.

1980 के दशक में सबसे पहले case tools का निर्माण किया गया था.

केस टूल्स को प्रस्तावित करने का मकसद सॉफ्टवेयर को विकसित करने में लगने वाले समय तथा cost (मूल्य) को कम करना था तथा सॉफ्टवेयर की क्वालिटी को बेहतर बनाना था.



### components of case tools:-

इस के components निम्नलिखित हैं.

- 1:- upper केस टूल्स
- 2:- lower केस टूल्स
- 3:- integrated केस टूल्स

**1:- upper case tools:-** upper केस टूल्स सॉफ्टवेयर डेवलपमेंट के पहले के phases में प्रयोग में आते हैं. ये tools प्लानिंग, एनालिसिस तथा डिजाइन फेज में use होते हैं.

**2:- lower case tools:-** ये case tools सॉफ्टवेयर डेवलपमेंट के बाद के फेजों implementation, testing, तथा maintenance में प्रयोग की जाती है.

**3:- integration case tools:-** इसे I-CASE भी कहते हैं. इन टूल्स का प्रयोग SDLC की सभी phases में किया जा सकता है.

### **types of case tools in hindi:-**

इसके प्रकार निम्नलिखित हैं:-

**1:- diagramming tools:-** इन tools का प्रयोग सिस्टम प्रोसेस, डेटा तथा कंट्रोल स्ट्रक्चर एवं सिस्टम कंपोनेंट्स को चित्रों के रूप में प्रस्तुत करने के लिए किया जाता है.

**2:- analysis tools:-** इन tools का प्रयोग डायग्राम, फॉर्मर्स तथा रिपोर्ट्स को एनालिसिस करने के लिए किया जाता है.

**3:- design tools:-** डिजाइन टूल्स का प्रयोग सॉफ्टवेयर डिजाइनरों के द्वारा सॉफ्टवेयर स्ट्रक्चर को डिजाइन करने के लिए किया जाता है.

**4:- process modeling tools:-** इन tools का प्रयोग सॉफ्टवेयर के प्रोसेस मॉडल को बनाने तथा बदलने के लिए किया जाता है.

**5:- project management tools:-** इन टूल्स का प्रयोग कोडिंग को करने के लिए किया जाता है. अर्थात ये tools सॉफ्टवेयर प्रोजेक्ट बनाने में सहायता करता है.

**6:- quality tools:-** सॉफ्टवेयर की क्वालिटी को बेहतर बनाने के लिए इन tools का प्रयोग किया जाता है.

**7:- maintenance tools:-** जब सॉफ्टवेर का निर्माण हो जाता है तब उसे maintain करने की जरूरत पड़ती है और समय के साथ साथ उसमें बदलाव भी किये जाते हैं. इस कार्य को करने के लिए maintenance tools का प्रयोग किया जाता है.

**8:- prototyping tools:-** इन टूल्स का प्रयोग सॉफ्टवेयर को नया look देने तथा उसे अच्छा style करने के लिए किया जाता है. इन tools में बहुत सारे ग्राफिक्स होते हैं जो कि यूजर इंटरफ़ेस तथा डिजाईन को अच्छा बना देते हैं.

**निवेदन:-** आपके लिए यह पोस्ट helpful रहो हो तो हमें कमेंट के द्वारा बताइए तथा इस पोस्ट को अपने दोस्तों के साथ share करें. धन्यवाद.

---

## what is COCOMO in hindi? software engineering

by yugal joshi - Saturday, June 25, 2016

<https://ehindistudy.com/2016/06/25/what-is-cocomo-in-hindi/>

**COCOMO model in hindi:-**

इस का पूरा नाम constructive cost model है इसको सन् 1981 में BOEHM ने प्रस्तावित किया था।

यह एक cost estimation मॉडल है तथा इसका प्रयोग सॉफ्टवेयर पैकेज की cost को evaluate करने के लिए किया जाता है।

इस मॉडल में effort equation का प्रयोग एक प्रोजेक्ट को विकसित करने के लिए person-month संख्या की जरूरत को estimate() करने के लिए किया जाता है।

BOEHM के अनुसार cost को estimate करने के लिए तीन stages का प्रयोग किया जाता है जो निम्नलिखित हैं:-

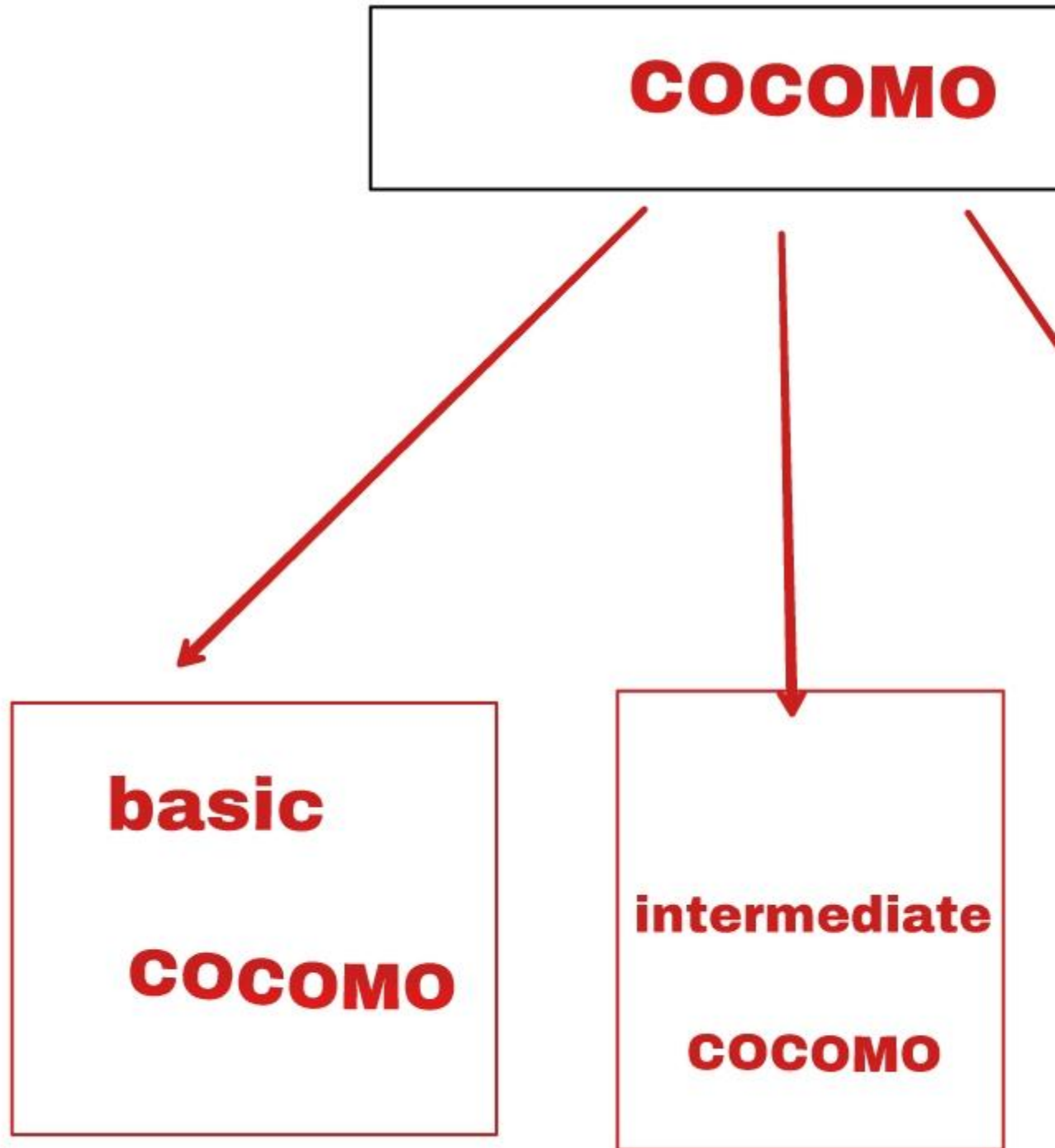


Fig:-phases of COCOMO

1:-Basic कोकोमो model:-



यह मॉडल स्थैतिक तथा single valued होता है जो कि सॉफ्टवेयर डेवलपमेंट effort तथा cost को function के program size की तरह compute करता है तथा program size जो है वह अनुमानित(estimated) lines of code(LOC) में व्यक्त होता है।

सामान्यतया इस मॉडल का प्रयोग छोटे तथा मध्यम आकर के सॉफ्टवेयर प्रोजेक्ट्स में किया जाता है।

इस मॉडल में cost को estimate करने के लिए निम्नलिखित सूत्र का प्रयोग किया जाता है:-

$$\text{Effort} = a_1 \times (\text{KLOC})^{a_2}$$

$$\text{Tdev} = b_1 \times (\text{Effort})^{b_2}$$

जहां;

\*KLOC सॉफ्टवेयर का अनुमानित आकर है।

\*  $a_1, a_2, b_1, b_2$  सॉफ्टवेयर की प्रत्येक category के लिए नियतांक(constant) है।

\* Tdev सॉफ्टवेयर को विकसित करने के लिए अनुमानित समय है।

\*Effort सॉफ्टवेयर को विकसित करने में लगा कुल effort है।

\*PM person-month है।

## 2:-Intermediate कोकोमो:-

यह मॉडल basic COCOMO का extension होता है।

Intermediate COCOMO सॉफ्टवेयर डेवलपमेंट effort को program size के फंक्शन तथा cost drivers के समूह की तरह कंप्यूट करता है। cost drivers प्रोजेक्ट में लगे समय तथा effort को निर्धारित करता है।

यह मॉडल basic मॉडल से बेहतर परिणाम देता है क्योंकि इसमें cost drivers का प्रयोग किया जाता है।

### **3:-Complete कोकोमो:-**

यह मॉडल intermediate COCOMO का extension होता है।

यह मॉडल इंटरमीडिएट मॉडल से भिन्न होता है क्योंकि यह प्रोजेक्ट के प्रत्येक Phase के लिए effort multipliers का प्रयोग करता है।

complete COCOMO में प्रत्येक subsystem की cost को अलग-अलग estimate किया जाता है। इस विधि के कारण त्रुटियाँ बहुत ही कम होती हैं।

basic तथा intermediate COCOMO की कमी यह है कि यह सॉफ्टवेयर प्रोजेक्ट को single homogeneous entity की तरह consider करता है। इस कमी को complete COCOMO दूर करता है।

complete COCOMO estimation को कैलकुलेट करने के लिए बहुत ही जटिल procedures का प्रयोग करता है।

अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं।

---

## **what is data flow diagram (DFD) in hindi?**

by yugal joshi - Monday, May 15, 2017

<https://ehindistudy.com/2017/05/15/data-flow-diagram-dfd-hindi/>

## Data flow diagram (DFD) in hindi:-

data flow diagram इनफार्मेशन सिस्टम से गुजरने वाले डेटा के flow का एक ग्राफिकल प्रस्तुतीकरण है.

जैसा कि इसके नाम से पता चल रहा है यह केवल डेटा (सूचना) के फ्लो, डेटा कहाँ से आया, यह कहाँ जायेगा, तथा यह कैसे स्टोर होगा इस पर केन्द्रित होता है.

data flow diagram का प्रयोग [सॉफ्टवेयर](#) सिस्टम के overview को बनाने के लिए किया जाता है.

data flow diagram जो है वह आने वाले (इनपुट) डेटा फ्लो, जाने वाले (आउटपुट) डेटा फ्लो तथा स्टोर किये हुए डेटा को डायग्राम अर्थात् ग्राफिकल रूप में प्रस्तुत करता है लेकिन DFD इसकी प्रोसेस के बारे में विस्तार से नहीं बताता है. लेकिन इसके लिए flow chart है. flow chart और DFD दोनों अलग अलग टॉपिक है.

### types of DFD in hindi:-

DFD दो प्रकार का होता है.

1:- लॉजिकल DFD

2:- फिजिकल DFD

**1:- logical DFD:-** लॉजिकल DFD बिज़नेस एक्टिविटी पर केन्द्रित रहता है अर्थात् यह सिस्टम में डेटा के फ्लो तथा सिस्टम के प्रोसेस पर केन्द्रित रहता है.

**2:- physical DFD:-** फिजिकल DFD इस बात पर केन्द्रित रहता है कि वास्तव में डेटा फ्लो सिस्टम में किस प्रकार implement (कार्यान्वित) हुआ है.

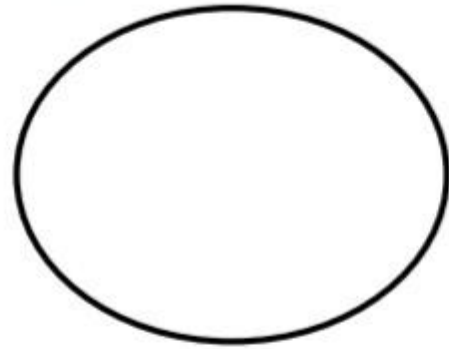
Data flow diagram का सबसे पहला प्रयोग 1970 के दशक में किया था तथा इसे larry constantine तथा Ed yourdon ने वर्णित किया था.

### DFD components:-

-----



External Entity



Process

DFD में निम्न चार मुख्य कंपोनेंट्स होते हैं.

- 1:- entities
- 2:- data storage
- 3:- processes
- 4:- data flow

**1:- entities:-** डेटा के source तथा destination को entities कहते हैं. entities को rectangle (आयत) के द्वारा प्रदर्शित किया जाता है.

**2:- data storage:-** ऐसी जगह जहाँ डेटा स्टोर होता है. इसे एक ऐसे आयत से प्रदर्शित किया जाता है जिसकी एक साइड नहीं होती या दोनों साइड नहीं होती है.

**3:- process:-** यह एक कार्य होता है जो कि सिस्टम के द्वारा किया जाता है. इसे circle (वृत्त) या round-edged rectangle (गोल-धारित आयत) के द्वारा प्रदर्शित किया जाता है.

**4:- data flow:-** यह डेटा के movement (गति) को दिखाता है. इसे arrow (तीर) द्वारा प्रदर्शित किया जाता है.

**निवेदन:-** आपको यह पोस्ट कैसी लगी हमें comment के माध्यम से बताएं तथा इसे अपने दोस्तों के साथ share करें. धन्यवाद.

---

# what is equivalence partitioning & boundary value analysis in hindi? software testing.

by yugal joshi - Tuesday, December 06, 2016

<https://ehindistudy.com/2016/12/06/equivalence-partitioning-boundary-value-analysis-hindi-software-testing/>

## Software testing techniques in hindi:-

यहाँ हम दो प्रकार के सॉफ्टवेर टेस्टिंग तकनीक के बारे में पढ़ेंगे जो निम्न है:-

- 1:-equivalence partitioning
- 2:- boundary value analysis

### 1:- equivalence partitioning in hindi:-

equivalence partitioning एक सॉफ्टवेर टेस्टिंग तकनीक है. इस तकनीक में सॉफ्टवेर यूनिट के इनपुट डेटा को equivalence classes (portitions) में विभाजित किया जाता है. equivalence classes का अर्थ है कि समान प्रकार का डेटा एक क्लास में विभाजित होगा और इस प्रकार अन्य समान डेटा समानता के आधार पर अलग-अलग classes में विभाजित होंगे.

इस तकनीक में हम प्रत्येक क्लास से केवल एक कंडीशन को ही टेस्ट करते हैं क्योंकि हमें पता होता है कि एक क्लास के सभी conditions एक जैसे ही होते हैं. अगर क्लास कि एक कंडीशन कार्य करती है तो उस क्लास की सभी conditions कार्य करेंगी. अगर क्लास कि एक टेस्ट condition कार्य नहीं करती है तो क्लास की कोई भी condition कार्य नहीं करेगी.

equivalence partitioning तकनीक का फायदा यह है कि इससे सॉफ्टवेर को टेस्ट करने में समय कम लगता है.

इसे टेस्टिंग के सभी levels जैसे:- यूनिट टेस्टिंग, security testing, alpha beta testing आदि पर प्रयुक्त किया जा सकता है.

**उदाहरण के लिये:-** माना कि एक दुकान में किसी ग्राहक को उसकी खरीद के हिसाब से डिस्काउंट दिया जाता है.

यदि खरीद ₹50 से ₹100 के मध्य की होती है तो कोई डिस्काउंट नहीं दिया जाता है, यदि खरीद ₹100 से ₹200 के मध्य की होती है तो 3% का डिस्काउंट दिया जाता है, यदि खरीद ₹200 से ₹300 के मध्य होती है तो उसे 8% का डिस्काउंट दिया जाता है, यदि खरीद 300 के उपर की होती है तो 12% डिस्काउंट दिया जाता है.

नीचे दर्शाया गया है कि हमारे पास चार valid class है तथा एक invalid class है.

INVALID CLASS	VALID CLASS (NO DISCOUNTS)	VALID CLASS (3%)	VALID CLASS (8%)
10	50-100	100-200	200-

equivalence partitioning को equivalence class partitioning भी कहते हैं.

## boundary value analysis in hindi (बाउंड्री वैल्यू एनालिसिस):-

boundary value analysis (BVA) एक ब्लैक बॉक्स टेस्टिंग तकनीक है जो कि class (partition) के मध्य की boundaries की टेस्टिंग पर आधारित होती है.

इसका प्रयोग इनपुट डेटा की boundaries में त्रुटियों को ढूँढने में किया जाता है.

boundary दो प्रकार की होती है एक lower boundary (जहाँ से सीमा(range) प्रारंभ होती है) और दूसरी (जहाँ सीमा समाप्त होती है). और प्रत्येक boundary की एक valid value तथा एक invalid value होती है.

test cases इन दोनों valid तथा invalid वैल्यूज के आधार पर डिजाइन किये जाते हैं तथा हम सामान्यतया प्रत्येक बाउंड्री से एक test case का चुनाव करते हैं.

**उदाहरण के लिये:-** माना कि college में एक सॉफ्टवेयर है जो कि valid user name तथा paasword पर ही work करता है तथा यह निम्नतम 6 तथा अधिकतम 10 कैरेक्टर ही accept करता है.

valid range 5-10, invalid range 5 या उससे कम, तथा invalid range 11 या 11 से ज्यादा.

INVALID CLASS  6 से कम	VALID CLASS  6-10
---------------------------------	----------------------------

BVA को range checking (रेंज चेकिंग) भी कहते हैं.

Equivalence partitioning तथा boundary value analysis एक दूसरे के साथ जुड़े हुए हैं तथा ये टेस्टिंग के सभी levels पर एक साथ प्रयोग किये जाते हैं.

**निवेदन:-** अगर आपको यह पोस्ट पसंद आयी है तो हमें comment के माध्यम से बताइये तथा इसे अपने दोस्तों के साथ share करें. धन्यवाद.

---

## What is Formal technical review in hindi

by yugal joshi - Thursday, July 07, 2016

<https://ehindistudy.com/2016/07/07/what-is-formal-technical-review-in-hindi/>

## FORMAL TECHNICAL REVIEW IN HINDI:-

FTR जो है वह [Software Quality Assurances\(SQA\)](#) का एक क्रियाकलाप है जिसे सॉफ्टवेयर इंजीनियरों के द्वारा परफॉर्म किया जाता है।

FTR के द्वारा सॉफ्टवेयर प्रोडक्ट में सम्भावित faults या defects को identify किया जाता है।

अगर defects को पहले ही identify कर लिया जाये तो बेहतर होगा क्योंकि बाद में इनको ढूँढना तथा सही कर पाना बहुत मुश्किल हो जाता है।

इसमें सॉफ्टवेयर प्रोडक्ट की क्वालिटी को बेहतर करने के लिए प्रोडक्ट को analyze किया जाता है।



### **FTR का उद्देश्य:-**

इसको निम्नलिखित उद्देश्य के लिए प्रयोग किया जाता है:-

- 1:-त्रुटियों को detect करने के लिए।
- 2:-सॉफ्टवेयर की जरूरतों को verify करने के लिए।
- 3:-यह देखने के लिए कि सॉफ्टवेयर सही तरीके से विकसित हुआ है या नहीं।
- 4:-यह सुनिश्चित करने के लिए कि सॉफ्टवेयर प्रोडक्ट को पहले से तय किये गए standards के अनुसार represent किया गया है।
- 5:-प्रोजेक्ट को और अधिक व्यवस्थित बनाने के लिए।

FTR के अंदर [walkthroughs](#), [inspections](#) तथा formal reviews इत्यादि आते हैं।

अंत में यही कहना चाहूंगा कि "कोई भी FTR तभी सफल हो सकता है जब उसे सही ढंग से plan, control और attend किया जाये।



अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं। आपके कोई suggestions या questions हैं तो आप comment के माध्यम से बताएँ।

---

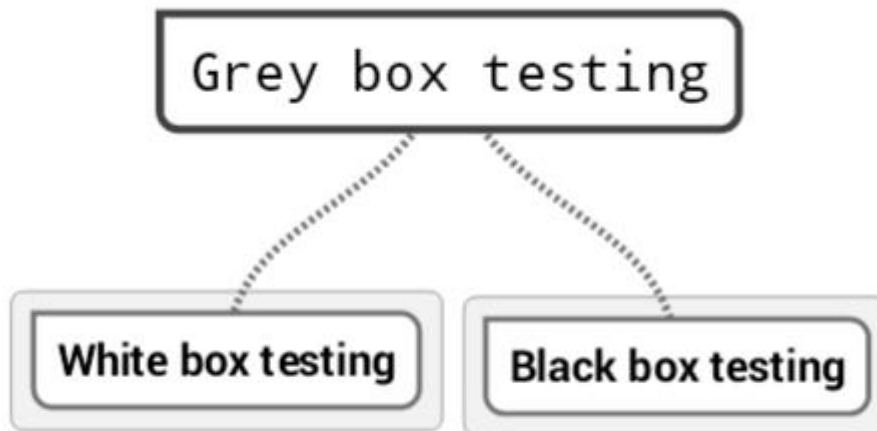
## what is gray box testing in hindi? software testing

by yugal joshi - Thursday, December 15, 2016

<https://ehindistudy.com/2016/12/15/gray-box-testing-hindi-software-testing/>

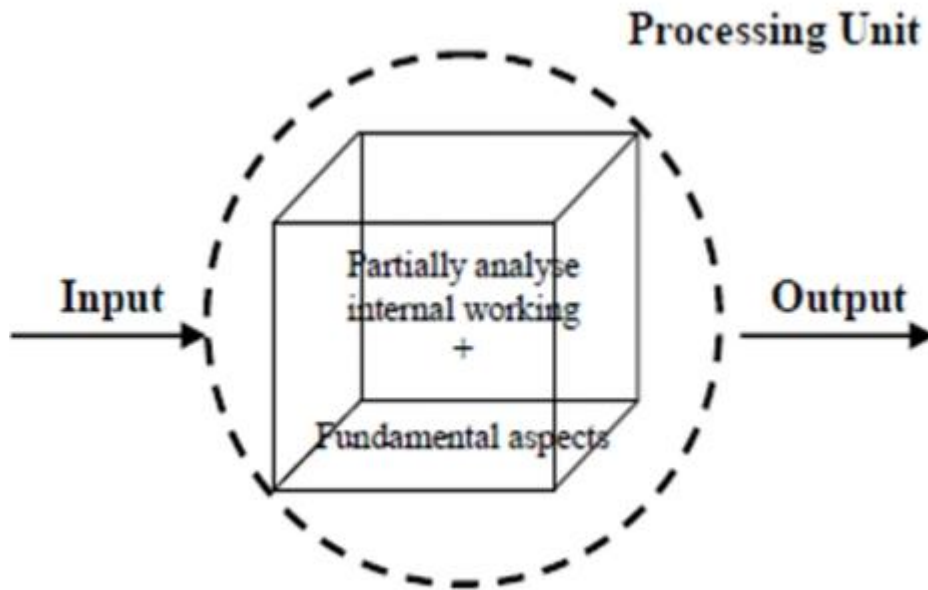
### Grey box testing (ग्रे बॉक्स टेस्टिंग) in hindi:-

grey box testing एक सॉफ्टवेर टेस्टिंग तकनीक है जो कि [white box](#) तथा [black box टेस्टिंग](#) का एक मिश्रण (combination) है.



black box टेस्टिंग में टेस्टर को सॉफ्टवेर के आंतरिक स्ट्रक्चर का ज्ञान नहीं होता है तथा white box टेस्टिंग में टेस्टर को सॉफ्टवेर के आंतरिक स्ट्रक्चर का ज्ञान होता है जबकि gray box testing में टेस्टर को आंतरिक तथा बाहरी दोनों भाग के functions का ज्ञान होता है.

इस टेस्टिंग का नाम gray इसीलिए पड़ा क्योंकि इसमें टेस्टर के लिए सॉफ्टवेयर अर्धपारदर्शी बॉक्स की तरह होता है जिसमें वह इसके अन्दर आंशिक रूप से देख सकता है.



grey box testing का लाभ यह है कि हम इसके द्वारा सॉफ्टवेयर के बाहरी भाग तथा आंतरिक भाग (कोडिंग) को टेस्ट कर सकते हैं.

इसका प्रयोग [इंटीग्रेशन टेस्टिंग](#) में किया जाता है.

इसका प्रयोग ज्यादातर इन्टरनेट पर आधारित सॉफ्टवेयर एप्लीकेशन को टेस्ट करने के लिए किया जाता है.

इस टेस्टिंग को **translucent testing** भी कहते हैं.

उदाहरण के लिए:- माना कि टेस्टर एक वेबसाइट को टेस्ट करता है अगर उसे उसमें broken links तथा अन्य अनुपयुक्त लिंक मिलते हैं तो वो उन्हें वेबसाइट के HTML भाग में जाकर उन्हें ठीक कर सकता है तथा उनकी उसी समय जांच कर सकता है.

**निवेदन:-** अगर आपको यह पोस्ट पसंद आयी है तो हमें comment के माध्यम से बताएं तथा इस पोस्ट को अपने दोस्तों के साथ share करें. धन्यवाद.

---

# What is Inspection in hindi?

by yugal joshi - Monday, July 04, 2016

<https://ehindistudy.com/2016/07/04/what-is-inspection-in-hindi/>

## Software testing Inspection in hindi:-

Inspection एक औपचारिक review है तथा इसमें तैयारी की आवश्यकता होती है।

इंस्पेक्शन को 1976 में M.E. Fagon ने सबसे पहले define किया था। इस कारण इसको Fagon Inspection भी कहते हैं।

इंस्पेक्शन के द्वारा त्रुटियों को detect तथा correct किया जाता है।

इसका नेतृत्व एक प्रशिक्षित Moderator के द्वारा किया जाता है। Moderator जो है वह author नहीं होता है। Inspection में एक सॉफ्टवेयर प्रोडक्ट को review के लिए चुना जाता है और इस सॉफ्टवेयर प्रोडक्ट के review के लिए एक inspection टीम एकत्रित होती है।

The image shows the words "Inspection" and "kya hai?" written in a red, hand-drawn, sketchy font. "Inspection" is on the top line and "kya hai?" is on the bottom line, both centered.

Fig:-Inspection

## Inspection का उद्देश्य:-

इसका निम्नलिखित उद्देश्य होता है:-

1:-इसका उद्देश्य सॉफ्टवेयर प्रोडक्ट में आने वाले defects को डिटेक्ट तथा करेक्ट करना है।

2:-सॉफ्टवेयर की क्वालिटी को बेहतर करना।

### **steps of inspection:-**

इंस्पेक्शन की प्रक्रिया में निम्नलिखित स्टेप्स होते हैं:-

1:-**Planning**:-इंस्पेक्शन की पूरी प्लानिंग moderator के द्वारा की जाती है और मॉडरेटर का कार्य सॉफ्टवेयर प्रोडक्ट का peer-review करना होता है।

2:-**Overview**:-इसमें author सॉफ्टवेयर प्रोडक्ट के background का वर्णन करता है।

3:-**Preparation**:-यह एक व्यक्तिगत क्रियाकलाप है ; इसमें प्रत्येक इंस्पेक्टर सॉफ्टवेयर प्रोडक्ट की जाँच करता है तथा सम्भावित defects को identify करता है। इसमें defects को केवल note किया जाता है लेकिन collect नहीं किया जाता है।

4:-**Meeting**:-यह मॉडरेटर के द्वारा संचालित की जाती है। मीटिंग में सॉफ्टवेयर के प्रत्येक हिस्से को क्रमानुसार पढ़ा जाता है। इसमें defects को collect कर लिया जाता है।

5:-**Rework**:-इसको author के द्वारा निष्पादित किया जाता है। मीटिंग के दौरान जो भी प्लान किया जाता है उसके हिसाब से सॉफ्टवेयर प्रोडक्ट में बदलाव किये जाते हैं।

6:-**Follow-up**:-इस में, author के द्वारा किये गए बदलावों की जाँच की जाती है यह देखने के लिए कि जो बदलाव किये हैं वे सही हैं या नहीं।

अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं। आपके कोई suggestions या questions हैं तो आप comment के माध्यम से बताएँ।

---

## **What is integration testing in hindi?**

by yugal joshi - Tuesday, July 19, 2016

<https://ehindistudy.com/2016/07/19/integration-testing-hindi/>

## Integration testing in hindi:-

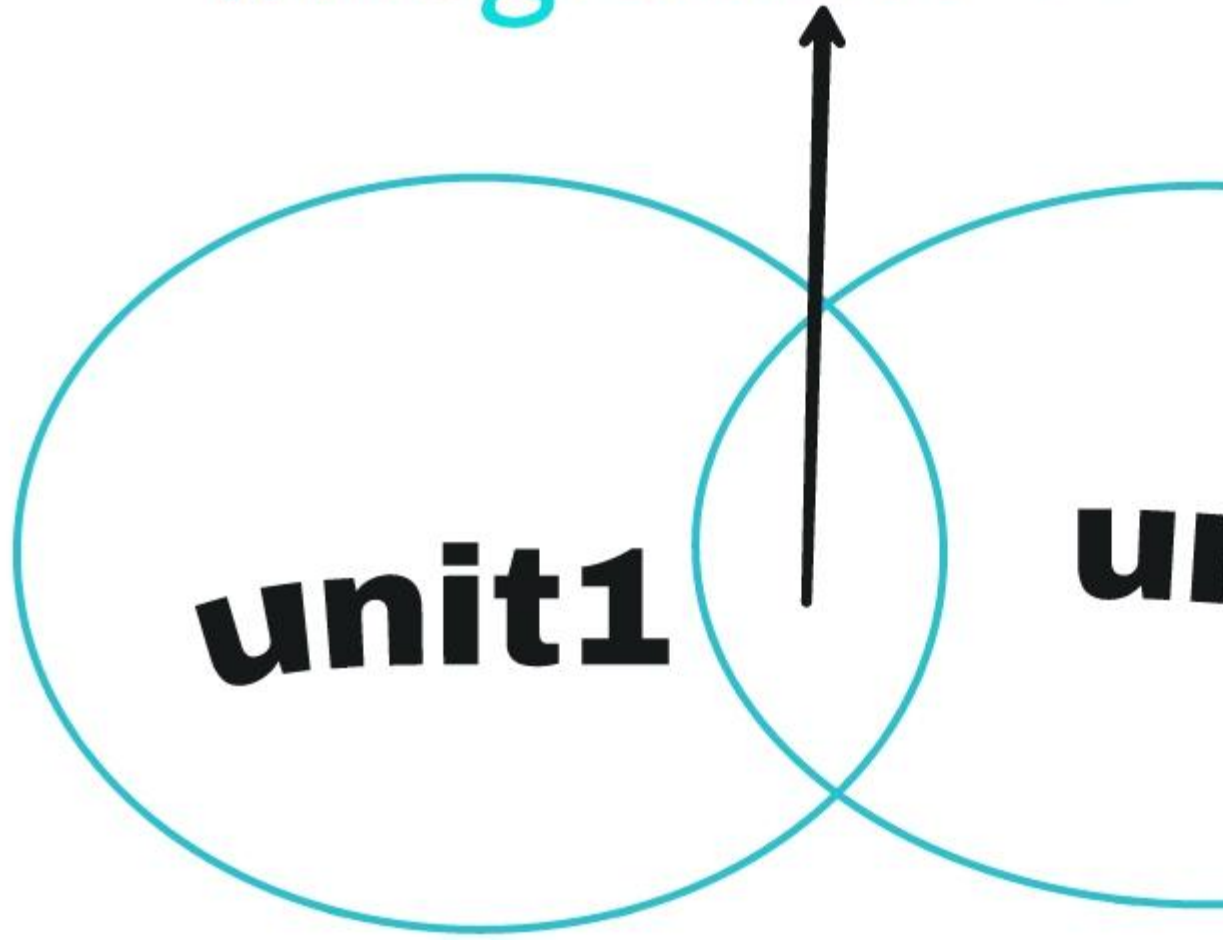
integration testing सॉफ्टवेर टेस्टिंग की एक विधि है जिसमें दो या दो से अधिक सॉफ्टवेर units या modules को एक साथ combine किया जाता है तथा एक समूह में उनकी टेस्टिंग की जाती है.

इस टेस्टिंग का मुख्य उद्देश्य integrated units के मध्य bugs तथा faults को खोजना है तथा units के मध्य कार्य, performance तथा विश्वसनीयता को identify करना है.

सामान्यतया intergation टेस्टिंग को एक integration टेस्टर के द्वारा perform किया जाता है.

यह टेस्टिंग यूनिट टेस्टिंग के बाद तथा validation टेस्टिंग के पहले की जाती है. जब यूनिट टेस्टिंग कर ली जाती है तो प्रत्येक unit को एक-एक करके integrate किया जाता है और यह क्रिया तब तक चलती है जब तक कि सारे units को integrate नहीं कर लिया जाएँ.

# integration test

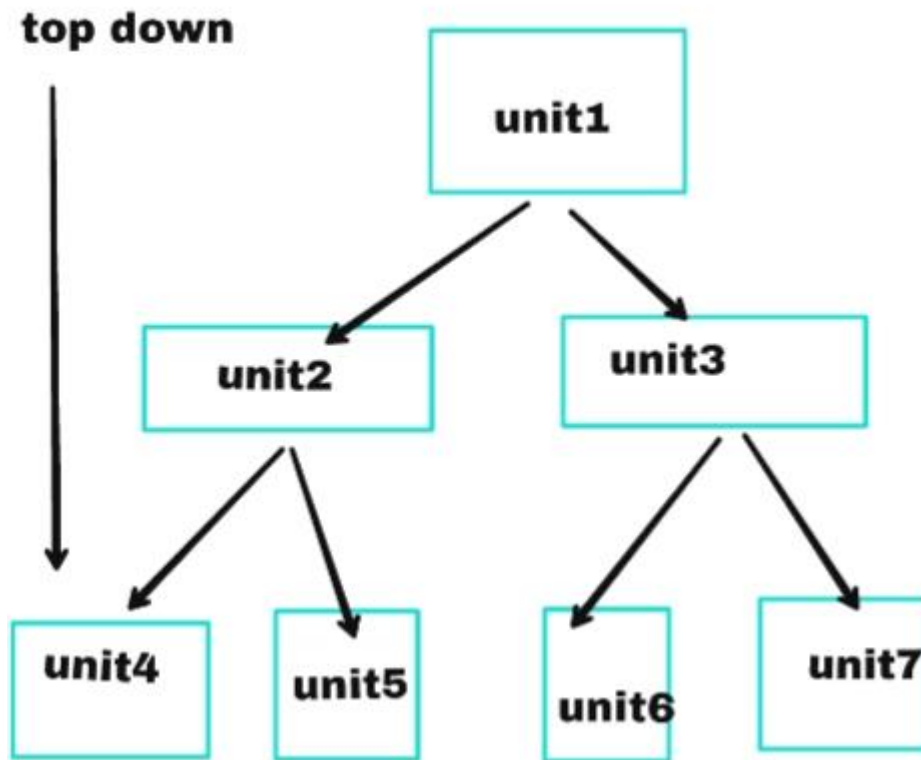


## integration testing approach in hindi:-

integration टेस्टिंग की बहुत सी approaches हैं उनमें से कुछ निम्नलिखित हैं:-

- 1:-Top-down approach
- 2:-Bottom-up approach
- 3:-Big bang approach

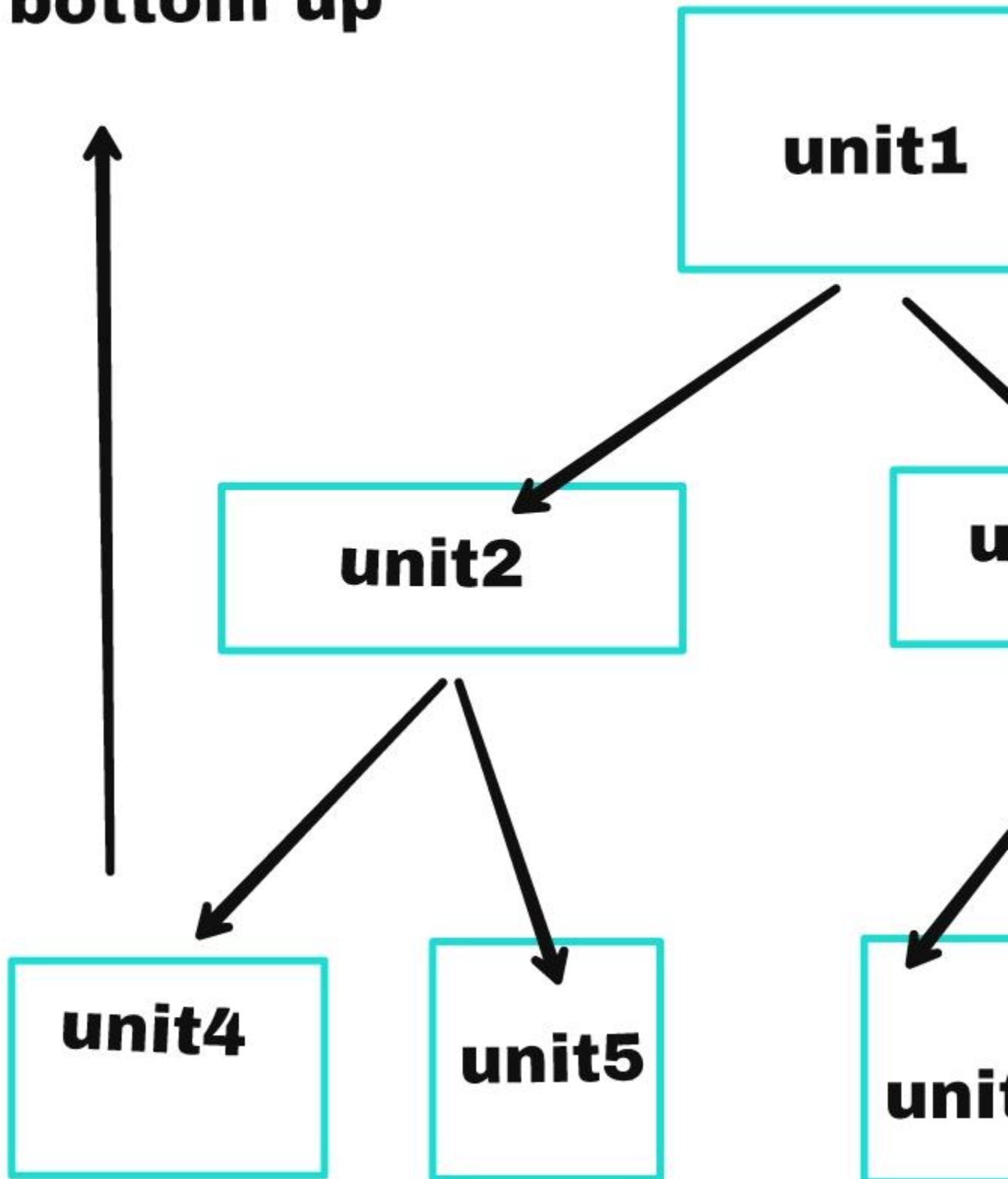
**1:-Top-down:-**इस integration टेस्टिंग में top level के integrated units को सबसे पहले test किया जाता है तथा उसके बाद उसके निचे वाले sub-units को test किया जाता है.



**2:-Bottom-up:-**इस integration टेस्टिंग में सबसे पहले bottom level के sub-units को test किया जाता है तथा उसके बाद ऊपर के मुख्य units को test किया जाता है.



**bottom up**



**3:-Big bang:-**इस प्रकार की टेस्टिंग में सभी units को एक साथ integrate कर लिया जाता है तथा इसके बाद एक समूह में सभी को test कर लिया जाता है. इस integration टेस्टिंग में किसी एक यूनिट को तब तक integrate नहीं किया जा सकता है जब तक कि सभी units तैयार ना हों.

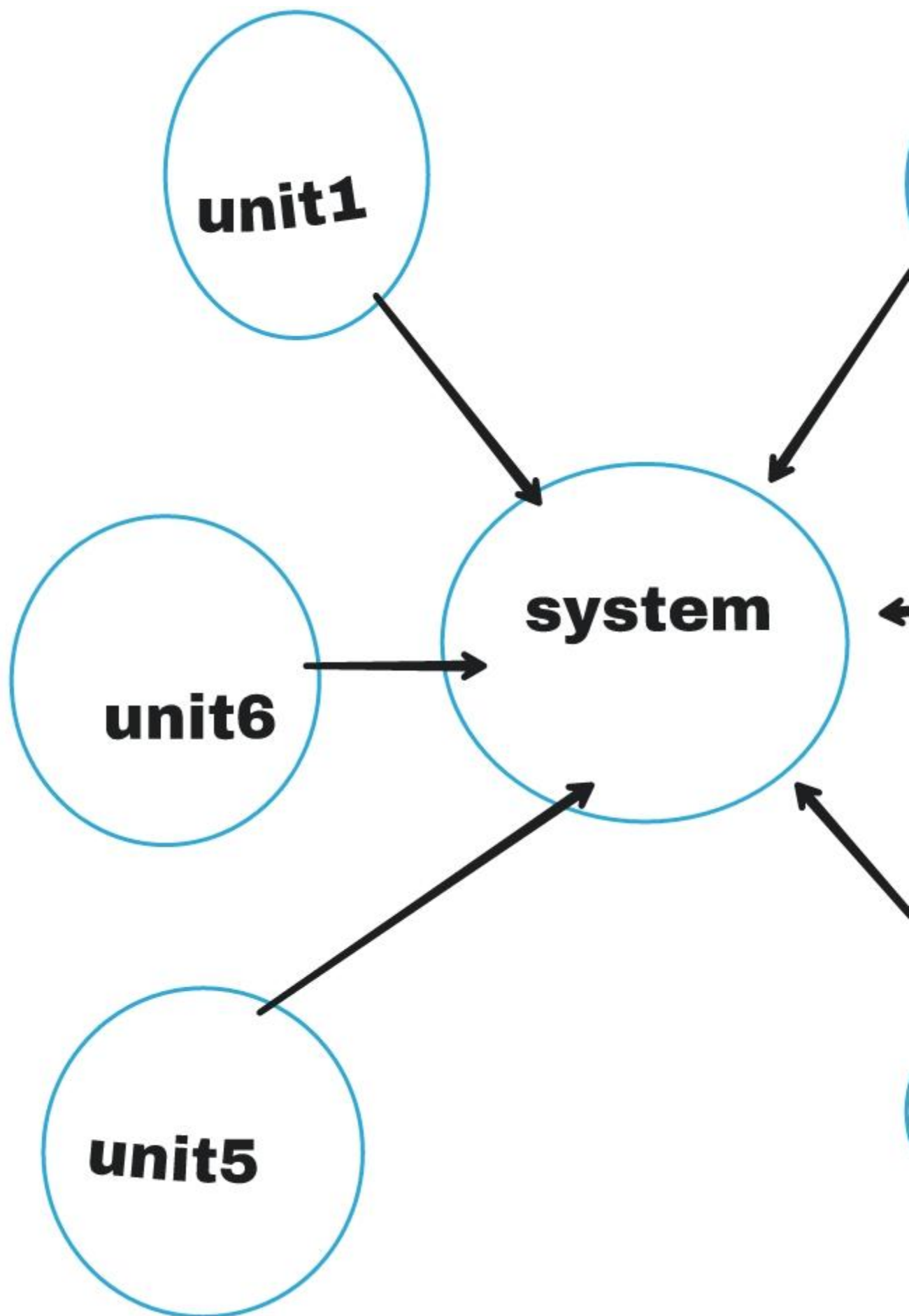


Fig:-big bang approach

यह टेस्टिंग बहुत ही risky होती है क्योंकि अगर हमने सही तरीके से डॉक्यूमेंटेशन नहीं किया तो इसमें failure होने का खतरा बढ़ जाता है.

इस टेस्टिंग का advantage यह है कि यह छोटे सिस्टम्स के लिए सुविधाजनक है. इसका disadvantage यह है कि इसमें faults तथा bugs को ढूँढ पाना बहुत ही मुश्किल होता है अगर कोई fault मिल भी जाता है तो fault के होने का मुख्य कारण का पता लगाना बहुत ही hard हो जाता है तथा इस टेस्टिंग में बहुत ही समय लग जाता है.

---

अगर यह पोस्ट आपके लिए helpful रही है तो हमें comment के माध्यम से बताएं तथा अपने दोस्तों के साथ share करें. धन्यवाद.

---

## What is Prototype model in hindi?

by yugal joshi - Friday, June 03, 2016

<https://ehindistudy.com/2016/06/03/what-is-prototype-model-in-hindi/>

### Prototype model in hindi:-

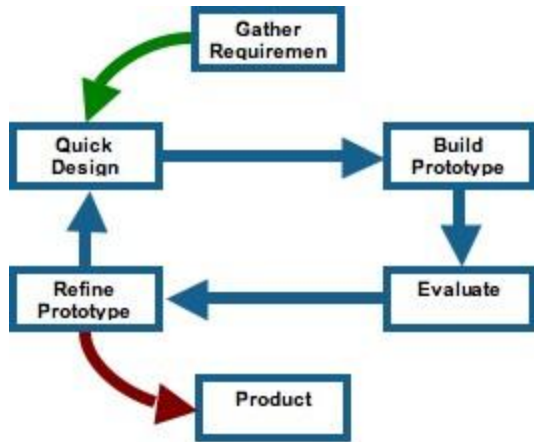
prototype model (प्रोटोटाइप मॉडल) एक ऐसी activity है जिसमें सॉफ्टवेयर एप्लीकेशन के prototypes का निर्माण किया जाता है। सबसे पहले प्रोटोटाइप का निर्माण किया जाता है फिर उस प्रोटोटाइप पर आधारित अंतिम product का निर्माण किया जाता है।

वाटरफॉल मॉडल की कमियों को दूर करने के लिए prototype model को विकसित किया गया था।

इस मॉडल का निर्माण तब किया जाता है जब हमें requirements का अच्छी तरह से पता नहीं होता है।

इस मॉडल की खासियत यह है कि इस मॉडल को दूसरे models के साथ भी प्रयोग में लाया जा सकता है और अकेले भी।

इस मॉडल में एक दिक्कत यह है कि end users अगर prototype मॉडल से संतुष्ट नहीं है तो दुबारा नया prototype मॉडल बनाया जाता है जिससे इस मॉडल में बहुत पैसा तथा समय लगता है।



## Phases in prototype model in hindi:-

prototype model में निम्नलिखित फेज होते हैं:-

**1:-Requirement gathering:-**prototype मॉडल का सबसे पहला स्टेप requirements को एकत्रित करना होता है, वैसे तो कस्टमर को requirements का कुछ खास पता नहीं होता लेकिन जो प्रमुख requirements हैं उनको विस्तार पूर्वक define कर लिया जाता है।

**2:-build the initial prototype:-**इस फेज में Initial prototype का निर्माण किया जाता है। इसमें कुछ बेसिक requirements को प्रदर्शित किया जाता है तथा यूज़र इंटरफ़ेस उपलब्ध किया जाता है।

**3:-review the prototype:-**जब प्रोटोटाइप का निर्माण पूरा हो जाता है तो end users या customer को इसे प्रस्तुत किया जाता है और उनसे इस प्रोटोटाइप के बारे में feedback लिया जाता है। इस फीडबैक का प्रयोग सिस्टम को और बेहतर बनाने में किया जाता है और प्रोटोटाइप में जो सम्भव परिवर्तन हो सकें वह किया जाता है।

**4:-Revise and improve the prototype:-**जब end users तथा customer से फीडबैक लिया जाता है तो फीडबैक के आधार पर प्रोटोटाइप को improve(बेहतर) किया जाता है। अगर कस्टमर प्रोटोटाइप से संतुष्ट नहीं है तो एक नये प्रोटोटाइप का निर्माण किया जाता है

और यह प्रक्रिया तब तक चलती रहती है जब तक कि कस्टमर को अपनी इच्छानुसार प्रोटोटाइप नहीं मिलता।

**nivedan:-** अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं। तथा comment के द्वारा बताइए धन्यवाद.

---

## what is RAD model in hindi?

by yugal joshi - Saturday, December 30, 2017

<https://ehindistudy.com/2017/12/30/rad-model-hindi/>

### **RAD model in hindi:-**

आज हम RAD model तथा इसके लाभ तथा हानियों के बारे में पढ़ेंगे तो चलिए शुरू करते हैं:-

RAD model का पूरा नाम rapid application development model है. RAD model की कार्यविधि जो है वह incremental या [waterfall model](#) की तरह ही सामान होती है. इसका प्रयोग छोटे projects के लिए किया जाता है. यदि प्रोजेक्ट बड़ा है तो उसे बहुत सारे छोटे छोटे projects में विभाजित कर लिया जाता है और इन छोटे projects की एक एक करके प्लानिंग की जाती है तथा इन्हें पूरा किया जाता है. इस प्रकार छोटे छोटे projects को पूरा करके बड़ा प्रोजेक्ट जल्दी तैयार हो जाता है.

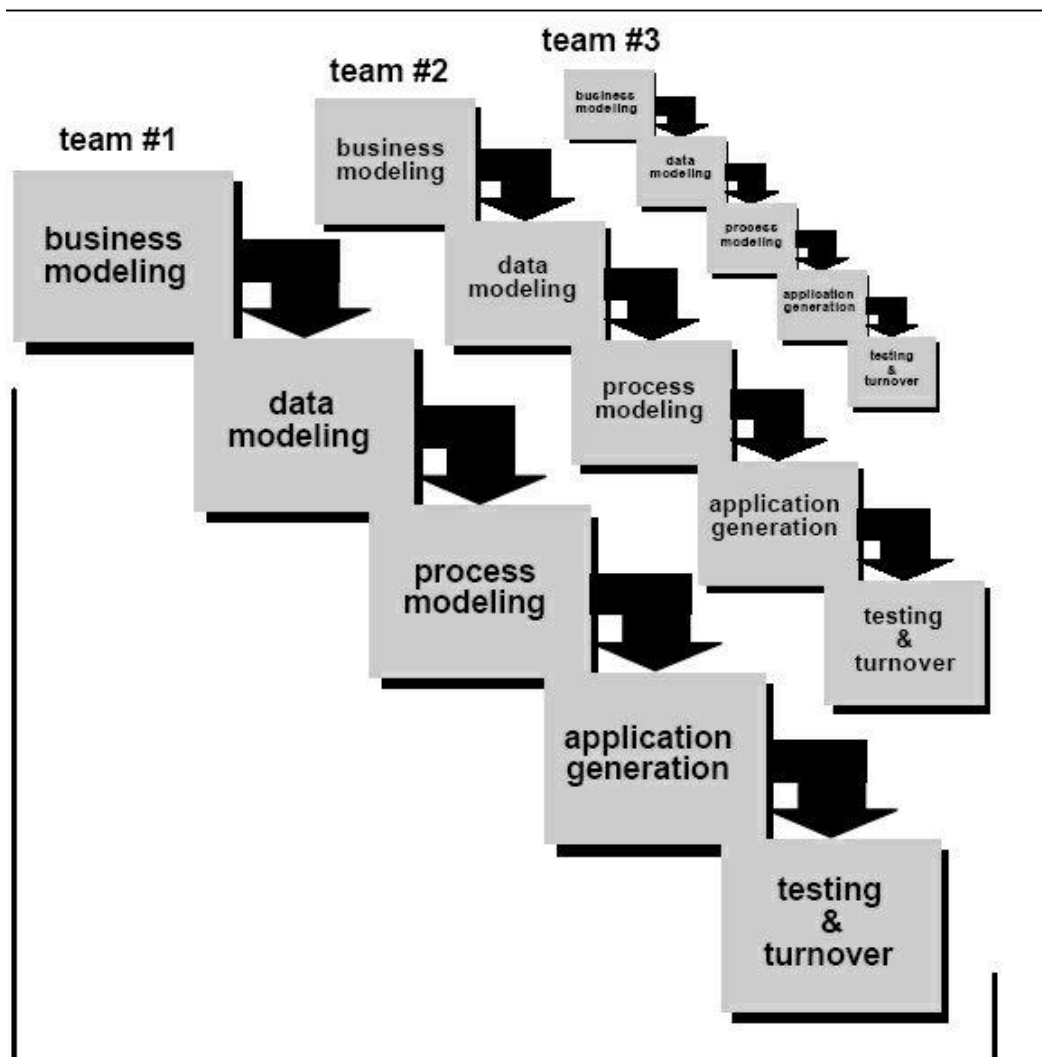
RAD model में, प्रोजेक्ट को दिए गये समय के अन्दर पूरा किया जाता है तथा प्रोजेक्ट को शुरू करने से पहले सारी requirements को इकट्ठा कर लिया जाता है.

RAD model में, प्रोजेक्ट बहुत ही तेजी से पूरा होता है तथा इसमें errors भी बहुत ही कम होती है.

RAD model का मुख्य उद्देश्य प्रोजेक्ट डेवलपमेंट में code, components, tools, processes को reuse करना है.

RAD model के निम्नलिखित phases होते हैं:-

- business modeling
- data modeling
- process modeling
- application generation
- testing and turnover



**1:- business modeling:-** इस फेज में, प्रोडक्ट के बिज़नेस मॉडल को विभिन्न बिज़नेस गतिविधियों द्वारा प्राप्त सूचना के आधार पर डिजाइन किया जाता है। डेवलपमेंट का कार्य शुरू करने से पहले business की पूरी जानकारी तथा functionality को भली भांति analyze किया जाता है।

**2:- data modeling:-** जब business modeling फेज समाप्त हो जाता है तो बिज़नेस मॉडलिंग से प्राप्त सूचना का प्रयोग data objects को डिफाइन करने के लिए किया जाता है।

**3:- process modeling:-** data modeling में जो डेटा ऑब्जेक्ट्स define किये थे उन्हें information flow को प्राप्त करने के लिए transform कर दिया जाता है जिससे कि बिज़नेस मॉडल के आधार पर कुछ विशेष business goals (लक्ष्यों) को पूरा किया जा सकें।

**4:- application generation:-** इस फेज में एप्लीकेशन को बनाया तथा कोडिंग की जाती है। तथा automated tools का प्रयोग सॉफ्टवेर को बनाने तथा कोडिंग करने के लिए किया जाता है। ये सब real time में होता है।

**5:- testing and turnover:-** इसमें सभी interfaces तथा components को टेस्ट किया जाता है। चूँकि इसमें prototype प्रत्येक चक्र में अलग से टेस्ट किये जाते हैं जिसके कारण RAD में testing time घट जाता है।

### **advantage of RAD model in hindi:-**

इसके लाभ निम्नलिखित हैं:-

**1:-** यह डेवलपमेंट में लगने वाले समय को कम कर देता है।

**2:-** इसमें components को reuse किया जाता है।

**3:-** यह flexible होता है तथा इसमें कोई भी बदलाव करना आसान है।



4:- इसे scripts की तरह ट्रान्सफर करना आसान है क्योंकि इसमें high level abstraction तथा intermediates codes का प्रयोग किया जाता है.

5:- इसमें defects बहुत ही कम होते हैं क्योंकि यह स्वभाव से ही [prototype](#) होता है.

6:- इसमें कम व्यक्तियों के साथ कम समय में productivity को बढ़ाया जा सकता है.

7:- यह cost effective होता है.

8:- छोटे प्रोजेक्ट्स के लिए यह suitable है.

### **disadvantage of RAD model:-**

इसके disadvantage निम्नलिखित हैं:-

1:- इसमें हमें highly skilled developers तथा designers की आवश्यकता होती है.

2:- इसे मैनेज करना बहुत मुश्किल होता है.

3:- यह लम्बे समय तक चलने वाले तथा बड़े प्रोजेक्ट्स के लिए सही नहीं है.

4:- इसमें प्रत्येक फेज के डेवलपमेंट के लिए क्लाइंट के फीडबैक की जरूरत पड़ती है.

5:- automated code generation बहुत महंगा होता है.

6:- यह मॉडल component based तथा scalable सिस्टम के लिए ही उपयुक्त है.

**निवेदन:-** अगर आपको यह पोस्ट पसंद आई है तो मुझे कमेंट के द्वारा बताएं तथा इस पोस्ट को अपने दोस्तों के साथ share करें.

---

## **what is recovery testing in hindi?**

by yugal joshi - Tuesday, September 13, 2016

<https://ehindistudy.com/2016/09/13/recovery-testing-hindi/>

### **recovery testing in hindi:-**

# RECO

# TES

रिकवरी टेस्टिंग को निम्न बिन्दुओं के आधार पर आसानी से समझा जा सकता है:-

1:-रिकवरी टेस्टिंग non-functional टेस्टिंग का एक प्रकार है.

2:-यह टेस्टिंग यह सुनिश्चित करने के लिए की जाती कि जब कोई एप्लीकेशन crash या fail हो जाती है तो वह कितनी जल्दी तथा कितने अच्छे तरीके से recover करती है.

3:-इस टेस्टिंग को करने के लिए एप्लीकेशन को खुद ही खराब या crash किया जाता है और फिर हर तरीके से इसमें रिकवरी टेस्टिंग की जाती है. जिससे टेस्टर को पता चलता है कि रिकवरी ठीक ढंग से हो पा रही है या नहीं.

निवेदन:-अगर आपके पास रिकवरी टेस्टिंग के बारे में और जानकारी है तो हमें comment के माध्यम से बताइए तथा इस पोस्ट को अपने दोस्तों के साथ share करें.

---

## what is security testing in hindi?

by yugal joshi - Tuesday, September 13, 2016

<https://ehindistudy.com/2016/09/13/security-testing-hindi/>

### security testing in hindi:-

security टेस्टिंग के द्वारा data तथा सॉफ्टवेर की functionality को सुरक्षित किया जाता है.

यह एक non-functional टेस्टिंग है अर्थात इसमें functional requirements की आवश्यकता नहीं होती है.

security टेस्टिंग एक प्रकार की सॉफ्टवेर टेस्टिंग है जिसमें यह निर्धारित किया जाता है कि क्या सिस्टम data तथा functionality को सम्भावित intruders(घुसपैठियों) से सुरक्षित रख पायेगा या नहीं?

security टेस्टिंग का लक्ष्य सिस्टम में उपस्थित threats को identify करना है तथा सभी संभावित risk को भी detect(चिन्हित) करना है.

# SECURITY

# TESTS

## **security services in hindi:-**

6 बेसिक security कॉन्सेप्ट्स निम्नलिखित हैं:-

- 1:-confidentially
- 2:-integrity
- 3:-authentication
- 4:-availability

5:-authorization

6:-non-repudiation

**Confidentially** का अर्थ है कि जो डेटा या सूचना है वह दुसरे व्यक्ति या पार्टी को disclose नहीं की जाएगी अर्थात केवल authorized यूजर ही सूचना को पढ़ पायेगा.

**Integrity** का अर्थ है कि जो सूचना या डेटा है वह केवल authorized यूजर के द्वारा ही modify, insert तथा delete की जा सकती है अर्थात unauthorized यूजर कोई मॉडिफिकेशन नहीं कर सकता है.

**authentication** में यूजर तथा प्रोडक्ट की वास्तविकता को confirm किया जाता है, तथा यह सुनिश्चित किया जाता है कि जो कंप्यूटर प्रोग्राम है वह विश्वास करने योग्य है.

**availability** यह सुनिश्चित करती है कि जो authorized यूजर है वह सूचना तथा resource को कभी-भी प्रयोग कर सकता है.

authorization का अर्थ है कि यूजर को किसी भी valid तथा requested ऑपरेशन के लिए allow किया जायेगा.

**non-repudiation** मैसेज transimission की गारंटी है अर्थात जब कोई व्यक्ति दुसरे को मैसेज करता है तथा दूसरा व्यक्ति मैसेज को recieve करता है तो बाद में दोनों व्यक्ति मैसेज के send तथा recieve होने से इंकार नहीं कर सकते हैं, अर्थात बाद में sender यह नहीं कह सकता है कि मैंने मैसेज send नहीं किया था तथा reciever यह नहीं कह सकता कि मैंने मैसेज recieve नहीं किया था.

**निवेदन:-**अगर आपको यह पोस्ट पसंद आयी है तो इसे अपने दोस्तों के साथ भी इसे share करें तथा अगर आप security टेस्टिंग के बारे में और कुछ जानते हो तो हमें comment के माध्यम से बताइए. धन्यवाद.

---

## What is software engineering in hindi & characteristics of a software in hindi

by yugal - Sunday, January 03, 2016

## सॉफ्टवेयर इंजीनियरिंग क्या है?:-

सॉफ्टवेयर इंजीनियरिंग का अर्थ एक ऐसी इंजीनियरिंग से है जिसमें कंप्यूटर सिस्टम तथा किसी अन्य इलेक्ट्रॉनिक डिवाइस के लिए सॉफ्टवेयर का निर्माण किया जाता है।

सॉफ्टवेयर इंजीनियर प्रोग्रामिंग तथा डिजाइनिंग करके इन सॉफ्टवेयरों तथा ऐप्लिकेशनों का निर्माण करते हैं।

## characteristics of software in hindi:-

एक अच्छे सॉफ्टवेयर के निम्नलिखित गुणधर्म होते हैं:-

**1:-Robustness:-** सॉफ्टवेयर में robustness होनी चाहिए जिससे कि सॉफ्टवेयर में defects तथा failure को detect किया जा सके।

**2:-Security:-** सॉफ्टवेयर में security बेहतर होनी चाहिए जिससे कि सॉफ्टवेयर सम्पूर्ण सुरक्षा बनी रहे और वह threats से सुरक्षित रहे।

**3:-Reliability:-** सॉफ्टवेयर reliable होना चाहिए अर्थात इसमें कोई defects नहीं होने चाहिए।

**4:-Flexibility:-** सॉफ्टवेयर flexible होना चाहिए अर्थात अगर सॉफ्टवेयर में कुछ बदलाव करने हैं तो वह आसानी से किये जा सकें।

**5:-Testability:-** सॉफ्टवेयर की टेस्टिंग आसानी से की जा सके।

**6:-Platform independent:-** सॉफ्टवेयर platform independent होना चाहिए अर्थात वह किसी भी सिस्टम में run हो सके।

keep learning.....

---

# what is software myths in hindi? software engineering

by yugal joshi - Wednesday, August 01, 2018

<https://ehindistudy.com/2018/08/01/software-myths-hindi/>

## Software myths in hindi

software development के लिए developer part पर dedication और understanding की आवश्यकता होती है। पर कभी कभी कुछ softwares को develop करते समय समस्या उत्पन्न होती है और यह समस्या software myths के कारण होती है।

"myth" का मतलब अफवा या भ्रम होता है , ऐसी बातें जो एक अफवा या भ्रम के जैसी होती हैं उसे myth कहते हैं। जब सॉफ्टवेयर को बनाया या develop किया जाता है तब developer को कुछ myth होते हैं , जिसकी वजह से software को develop करने में समस्या उत्पन्न होती है। software myths developer के दिमाग में झूठी मान्यता और भ्रम को उत्पन्न करती हैं। पर यह सिर्फ myth होता है reality या वास्तविकता कुछ और होती है।

यहाँ हम software development के myth और reality की तुलना करेंगे की software को develop करते समय manager , user and developers को क्या myth होता है जबकि reality कुछ और होती है। जिसकी वजह से software को आसानी से good quality का , काम budget में , और कम समय में software बना सकता है।

Software myths तीन प्रकार के होते हैं -

6. **Management myth**
7. **User myth**
8. **Developer myth**



**1:- Management myth :-** manager वह होते हैं जिनके ऊपर software को develop करने की जिम्मेदारी होती है। software development में manager को सॉफ्टवेयर की बेहतर quality , budget , समय की बाधाओं व कई बातों का pressure होता है । software को develop करने के लिए manager को कुछ myth होते हैं -

**a) information :-**

**Myth -** manager को यह myth होती है कि किसी भी software को बनाने के लिए developer के पास complete information होनी चाइये और यह information उन्हें manual से मिलता है जिसमें software को बनाने का procedure , principal व standard दिया हुआ होता है।

**Reality -** reality यह होती है कि software को थोड़ी सी जानकारी के साथ भी develop करना शुरू कर सकते हैं। अक्सर organization के standard manual समय के साथ पुराने हो जाते हैं। और जो developer होते हैं वह organization के manual को नहीं जानते हैं जिसकी वजह से manual के अनुसार काम करने से software delivery time बढ़ते जाती है इसलिए developer manual का उपयोग बहुत कम करते हैं।

**b) Scheduling :-**

**Myth -** जब software को deliver करने का समय निश्चित समय किये गए समय से ज्यादा होने लगता है तब कुछ companies व organization के manager के द्वारा यह सोचा जाता है की software को develop करने के लिए जितने ज्यादा developers होंगे software उतना ही जल्दी develop होगा।

**Reality -** पर reality यह होती है कि कुछ developer नए या fresher भी होते हैं जिन्हें software को समझने व develop करने में समय लगता जिससे software delivery time और ज्यादा बढ़ने लगता है।

### c) Outsourcing :-

**Myth** - जब कोई manager software को develop करने के लिए किसी third person को outsource ( ठेका देना ) देता है तब उसे ऐसा लगता है कि अब वह आराम कर सकता है उसे देख रेख करने की कोई जरूरत नहीं है। उनका software third person द्वारा बना दिया जायेगा।

**Reality** - पर ऐसा नहीं है reality यह होती है कि जब manager software development के लिए किसी third person को software देता है तब भी उसे देखरेख करनी पड़ती है उस third person के द्वारा ठीक से कम किया जा रहा है या नहीं। क्योंकि यदि third person ठीक से कम नहीं करेगा तो software में internal problem आ सकती है जिसके कारण बाद में organization को भुगतना पड़ सकता है।

**2:- User myth :-** ज्यादातर users को software के बारे में गलतफेमी होती है क्योंकि manager and developers user की गलतफैमि को दूर करने की कोशिश नहीं करते हैं जिसके वजह से user कभी भी software से सन्तुष्ट नहीं हो पाता है।

software को लेकर user के कुछ software myths -

#### a) requirements

**Myth** - user को ऐसा लगता है कि शुरुआत में थोड़ी सी requirement बताना काफी है software development के लिए बाद में बाकि पूरी requirement बाद में चरण में बता देंगे।

**Reality** - लेकिन software development के लिए शुरुआत में ही complete requirement बताना पड़ता है क्योंकि बाद में requirement बताने से extra resources and time लगता है जिसकी वजह से software का cost भी बढ़ जाता है।

## **b) Software flexibility:-**

**Myth** - user को लगता है कि software एक जैसा बन गया उसके बाद समय के साथ उसमें परिवर्तन नहीं किया जा सकता है।

**Reality** - जबकि वास्तविकता में ऐसा नहीं है software flexible होता है उसमें समय के साथ और जरूरत के अनुसार भविष्य में परिवर्तन किया जा सकता है।

**3:- Developer myth :-** software development को लेकर developers को कुछ software myths होते हैं -

### **a) Software deliver -**

**Myth** - developer को इस बात का भ्रम होता है कि software के coding के बाद software complete हो गया और software delivery के बाद उसका काम खत्म हो गया।

**Reality** - developer को software के delivery के बाद भी काम करना पड़ता है। developer का काम खत्म नहीं होता है। developer को software delivery के बाद भी maintenance का काम करना पड़ता है।

### **b) Project quality -**

**Myth** - developer यह सोचता है कि यदि project के लिए product का quality अच्छा होगा तो software भी अच्छा बनेगा।

**Reality** - यह जरूरी नहीं होता है कि software development के लिए जो अच्छी quality का product use किया जा रहा है उससे software भी अच्छा बने। software को अच्छा बनाने के लिए dedication और understanding की जरूरत होती है।

### c) Documentation section -

**Myth** - developer को इस बात का भ्रम होता है कि [software requirement and specification \(SRS\)](#) बनाने की जरूरत नहीं है। यह software development को slow कर देगा।

**Reality** - software engineering अच्छे quality के software create करती है। और documentation section software की quality को और बेहतर बनाता है। documentation की वजह से software पर दुबारा काम करने व maintenance में सुविधा होती है।

निवेदन:- आपको software myths की पोस्ट अच्छी लगी हो तो इसे अपने दोस्तों के अवश्य share करें. धन्यवाद.

---

## what is Software quality in hindi?

by yugal joshi - Sunday, June 26, 2016

<https://ehindistudy.com/2016/06/26/what-is-software-quality-in-hindi/>

### software quality in hindi:-

सॉफ्टवेयर क्वालिटी से तात्पर्य है कि सॉफ्टवेयर error free(त्रुटि मुक्त), आवश्यकताओं को पूरा करने वाला, समय पर deliver होने वाला तथा बजट के अंदर आने वाला होना चाहिए और उसकी performance उच्च कोटि की होनी चाहिए।

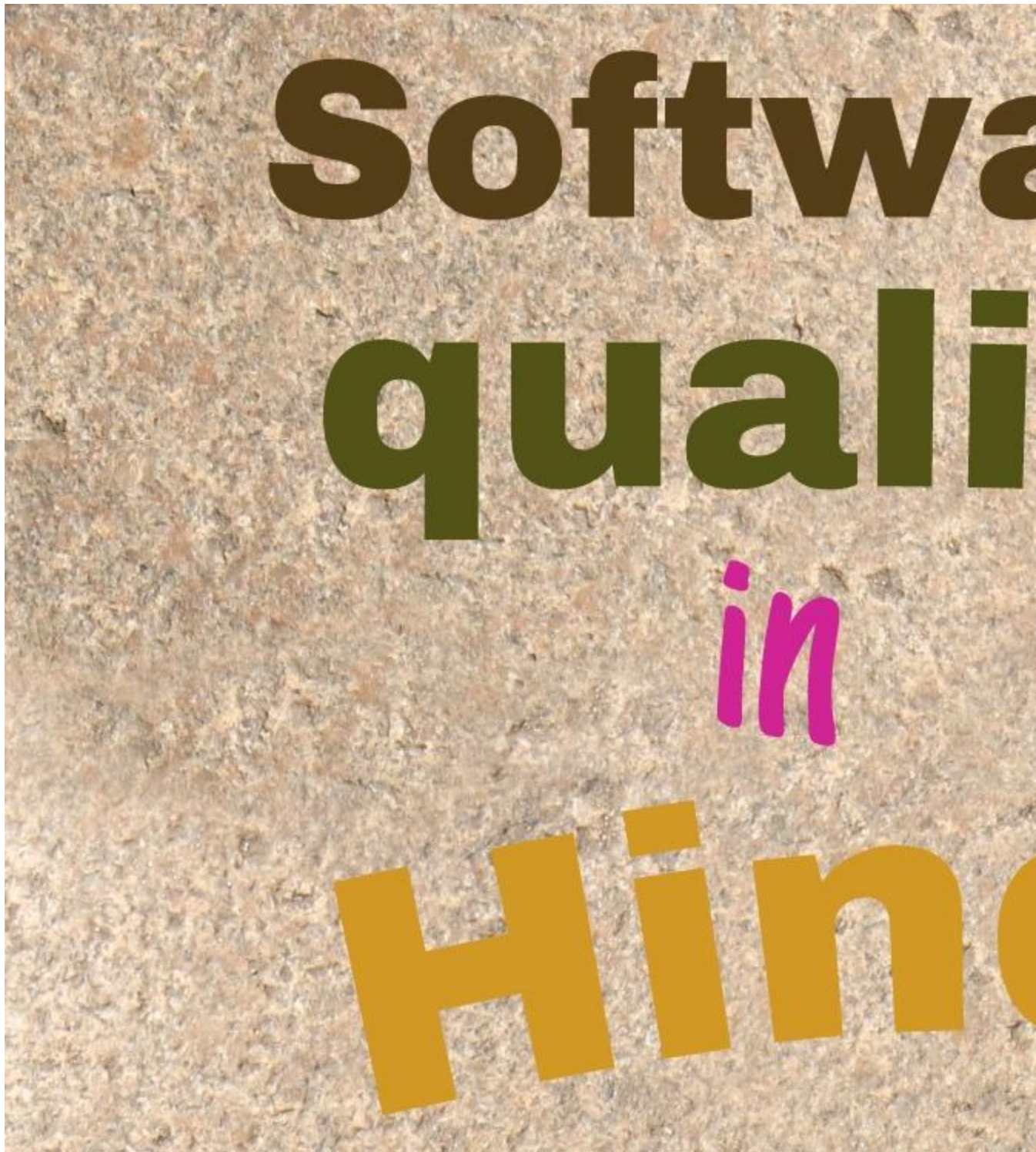


Fig:-software quality in hindi

सॉफ्टवेयर क्वालिटी सॉफ्टवेयर के external तथा internal क्वालिटी पर आधारित होता है।

external क्वालिटी का अर्थ है कि जब कोई यूजर सॉफ्टवेयर को चलाता है तो वह क्या experience करता है?

जबकि internal क्वालिटी ऐसे पहलू पर आधारित होता है जो कि code-dependent होता है और ये कोड end-users को दिखाई नहीं देता है इनका coding से कोई भी मतलब नहीं होता है।

सॉफ्टवेयर की external तथा internal क्वालिटी निम्नलिखित हैं:-

**1:-Ease-of-use:-**सॉफ्टवेयर को आसानी से चलाया जा सके अर्थात् सॉफ्टवेयर का यूजर-इंटरफ़ेस बेहतर होना चाहिए।

**2:-Design:-**सॉफ्टवेयर का डिज़ाइन अच्छा होना चाहिए क्योंकि यूजर सॉफ्टवेयर के प्रति तभी आकर्षित होगा जब उसका डिज़ाइन बेहतर होगा।

**3:-Speed(गति):-**सॉफ्टवेयर को कम समय गवांये अपनी सर्विस provide करनी चाहिए अर्थात् इसकी गति तेज होनी चाहिए।

**4:-Security(सुरक्षा):-**सॉफ्टवेयर secure होना चाहिए अर्थात् सॉफ्टवेयर में उपस्थित डेटा को कौन-कौन देख तथा प्रोसेस कर सकता है? अगर कोई इंटरनेट पर आधारित सॉफ्टवेयर है तो उसमें encryption तथा decryption होना चाहिए।

**5:-Error-free:-**सॉफ्टवेयर त्रुटियों से मुक्त होना चाहिए क्योंकि अगर सॉफ्टवेयर में errors होंगे तो सॉफ्टवेयर की reliability तथा security कम हो जायेगी जिससे सॉफ्टवेयर fail हो जायेगा।

**6:-Portable:-**सॉफ्टवेयर portable होना चाहिए अर्थात् सॉफ्टवेयर दूसरे environment पर भी आसानी से run होना चाहिए।

**7:-Testability:-**Testability का अर्थ है कि सॉफ्टवेयर को आसानी से टेस्ट किया जा सके।

**8:-Maintainability:-**इसका अर्थ है कि अगर हमें सॉफ्टवेयर में कोई बदलाव करने हो तो हमें उसे आसानी से कर सके।



**9:-Readability:-**सॉफ्टवेयर का source-कोड readable होना चाहिए अर्थात् जिसको आसानी से पढ़ा जा सके।

**10:-Reliability:-**सॉफ्टवेयर reliable होना चाहिए अर्थात् जो सॉफ्टवेयर है वह अपनी performance पर स्थिर होना चाहिए।

**11:-Functionality:-** अर्थात् जो सॉफ्टवेयर है उसमें यूजर के जरूरत के आधार पर functions होने चाहिए जिससे यूजर को उसमें कार्य करने में आसानी हो।

सॉफ्टवेयर क्वालिटी को measure करने के बहुत से criteria हैं और प्रत्येक पहलू की अपनी-अपनी importance होती है।

## **SQA in hindi:-**

SQA का पूरा नाम software quality assurance है। SQA ऐसी क्रियाकलापों का एक समूह है जो यह सुनिश्चित करता है कि विकसित किये गए सॉफ्टवेयर में क्वालिटी उपस्थित है।

SQA सॉफ्टवेयर डेवलपमेंट लाइफ-साइकिल(SDLC) के अंदर होने वाली प्रक्रिया है जो कि सॉफ्टवेयर क्वालिटी के लिए प्रयोग किये जाने वाली methods तथा process की समयानुसार जांच करता है।

अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं।

---

## **what is software risks in hindi & types of software risks in hindi.**

by yugal joshi - Sunday, October 16, 2016

<https://ehindistudy.com/2016/10/16/software-risks-hindi-types-software-risks-hindi/>

## **What is software risks in hindi:-**

सबसे पहले हम यह समझेंगे कि risk क्या है?

“RISK भविष्य की एक संभावित(अनिश्चित) घटना है जिसमें हमें नेगेटिव तथा अवांछनीय(undesirable) परिणाम मिल सकते हैं. जिसकी प्रायिकता 0 से 100% तक हो सकती है लेकिन यह प्रायिकता है सत्यता नहीं.”



## types of software risks in hindi(सॉफ्टवेर रिस्क के प्रकार):-

software risk निम्नलिखित प्रकार के होते हैं:-

- 1:-management risks
- 2:-technical risks
- 3:-financial or budget risks
- 4:-schedule risks
- 5:-contractual & legal risks
- 6:-operational risks

**1:-management risks:-**इस risk के मुख्य कारण निम्नलिखित हैं:-

- \*अच्छी तरीके से प्लानिंग ना हो पाना.
- \*मैनेजमेंट का अनुभव निम्नस्तरीय होना तथा प्रशिक्षण की कमी.
- \*सही ढंग से कम्युनिकेशन ना हो पाना.
- \*organisation के मुद्दों का हल ना मिल पाना.
- \*प्राधिकरण के अभाव का होना.

**2:-Technical risks:-**इस risk के मुख्य कारण निम्नलिखित हैं:-



- \*जरूरतों में लगातार परिवर्तन होते रहना.
- \*नयी तकनीक का उपलब्ध न होना.
- \*कोडिंग का उच्चस्तरीय भाषा में ना होना.
- \*सॉफ्टवेर की कार्यछमता अच्छी न होना.

**3:-Budget risk:-** किसी भी प्रोजेक्ट में पैसा एक बहुत ही महत्वपूर्ण factor होता है इसलिए इसमें risks भी बहुत होता है. इस risk के होने के मुख्य कारण निम्नलिखित हैं:-

- \*अनुमानित बजट के अंदर सॉफ्टवेर का पूरा ना होना.
- \*संसाधनों की लागत में लगातार वृद्धि होना.
- \*सॉफ्टवेर प्रोजेक्ट का बढ़ जाना.
- \*सॉफ्टवेर प्रोजेक्ट में देरी का होना.

**4:-schedule risks:-** इस risk के मुख्य कारण निम्नलिखित हैं:-

- \*अनुमानित समय के अंदर प्रोजेक्ट का पूरा ना होना.
- \*संसाधनों का आबंटन गलत हो जाना.
- \*अचानक से प्रोजेक्ट का विस्तार हो जाना.
- \*संसाधनों को सही ढंग से track ना कर पाना.

**5:-Contractual तथा legal risks:-** किसी भी प्रोजेक्ट में legal तथा contractual risks भी होते हैं जो निम्न हैं:-

- \*सरकारी नियम तथा अधिनियम के अंदर सॉफ्टवेर का निर्माण ना होना.
- \*सॉफ्टवेर की वारंटी के अन्दर ही सॉफ्टवेर में कोई खराबी आ जाना.
- \*सॉफ्टवेर की रिक्वायरमेंट्स में बदलाव होना.
- \*सुरक्षा तथा स्वास्थ्य के मुद्दे आदि.

**6:-operational risks:-** इस risk के मुख्य कारण निम्नलिखित हैं:-

- \*प्रोसेस का सही ढंग से लागू ना हो पाना.
- \*team spirit(सहयोग भावना) की कमी का होना.
- \*अपनी जिम्मेदारियों में स्पष्टता की कमी का होना अर्थात अपनी जिम्मेदारियों का सही ढंग से पालन ना करना.

\*अभ्यास की कमी का होना.

\*सही ढंग से कम्युनिकेशन का ना हो पाना अर्थात टीम में कम्युनिकेशन का अभाव.

**7:-अन्य risks:-** उपर दिए गये risks के अलावा और भी risks होते हैं जो निम्नलिखित हैं:-

\*सरकार की policy में बदलाव का हो जाना.

\*टूल्स का पर्याप्त न होना.

\*सुविधाओं का पर्याप्त ना होना.

\*कंप्यूटर रिसोर्स का उपलब्ध ना होना.

\*contracts का ना मिलना या मिले हुए contracts खो देना.

**निवेदन:-** अगर आपको यह पोस्ट पसंद आयी है तो हमें comment के माध्यम से बताइये तथा इस पोस्ट को अपने दोस्तों के साथ share करें. धन्यवाद.

---

## what is SRS in hindi & characteristics of SRS in hindi

by yugal joshi - Saturday, April 22, 2017

<https://ehindistudy.com/2017/04/22/srs-hindi-characteristics-srs-hindi/>

## what is SRS and characteristics of SRS in hindi:-

### SRS (SOFTWARE REQUIREMENT SPECIFICATION) in hindi:-

software requirement specification जो है वह किसी सिस्टम या [सॉफ्टवेर](#) एप्लीकेशन की जरूरतों का एक पूरा डॉक्यूमेंट या विवरण होता है.

अर्थात् दुसरे शब्दों में कहें तो, “SRS एक डॉक्यूमेंट होता है जो कि यह describe करता है कि सॉफ्टवेयर के features क्या होंगे तथा उसका behaviour क्या होगा अर्थात् वह किस प्रकार परफॉर्म करेगा.”

SRS का फायदा यह है कि इससे developers को सॉफ्टवेयर को विकसित करने में कम समय तथा मेहनत लगती है.

SRS जो है वह सॉफ्टवेयर के layout की तरह होता है जिसकी यूजर समीक्षा कर सकता है तथा देख सकता है कि वह (SRS) उसकी जरूरतों के हिसाब से बना है या नहीं”

हमने software requirement specification के बारे में जान लिया कि वह क्या है अब उसकी विशेषताओं को पढ़ते हैं.

**characteristics of SRS in hindi (विशेषताएं):-** इस की विशेषताएं निम्नलिखित हैं:-

1:- accurate 2:- clear 3:- complete 4:- consistent 5:- verifiable 6:- modifiable 7:- traceable  
8:- testable 9:- unambiguous 10:-correct

**1:- complete (संपूर्ण):-** SRS जो है वह संपूर्ण होना चाहिए अर्थात् जो भी सॉफ्टवेयर की जरूरतें हैं वह सभी SRS में उल्लेखित होनी चाहिए.

**2:- correct (उचित):-** यह correct होना चाहिए अर्थात् जो कस्टमर की जरूरतें हैं उसके हिसाब होना चाहिए.

**3:- clear (स्पष्ट):-** यह स्पष्ट होना चाहिए. सॉफ्टवेयर की आवश्यकताओं को स्पष्ट रूप से वर्णित (declared) होना चाहिए.

**4:- accurate (परिशुद्ध):-** इसमें accuracy होनी चाहिए. अगर यह ही accurate नहीं होगा तो सॉफ्टवेयर का निर्माण नहीं हो सकता.

**5:- consistent (निरंतर):-** इसको शुरुवात से लेकर अंत तक consistent (निरंतर) होना चाहिए जिससे कि यूजर आसानी से रिक्वायरमेंट्स को समझ सकें. और consistency तभी प्राप्त की जा सकती है जब दो रिक्वायरमेंट्स के मध्य कोई विरोधाभास ना हो.

**6:- verifiable (सत्यापनीय):-** यह verifiable (सत्यापन योग्य) होना चाहिए. जो experts तथा testers होते हैं उनके द्वारा requirements को verify किया जाता है.

**7:- modifiable (परिवर्त्य):-** SRS का जो डॉक्यूमेंट होता है उसमें specify की गयी सारी रिक्वायरमेंट्स मॉडिफाई करने योग्य होनी चाहिये. यह तभी हो सकता है जब SRS का स्ट्रक्चर संतुलित हों.

**8:- Traceable:-** यह traceable होना चाहिए अर्थात् इसमें प्रत्येक requirement अलग अलग प्रकार से identify होनी चाहिए. प्रत्येक requirement की अपनी अलग एक पहचान होनी चाहिए.

**9:- testable (परिक्षण योग्य):-** यह testable होना चाहिए अर्थात् इसे किसी भी प्रकार से टेस्ट करने के योग्य होना चाहिए.

**10:- unambiguous (अमिश्रित):-** यह केवल तब unambiguous हो सकता है जब सभी requirement का केवल एक ही अर्थ हों. अर्थात् केवल एक ही interpretation (व्याख्या) हो.

**निवेदन:-** आपको यह पोस्ट कैसी लगी हमें comment के द्वारा बताइये तथा इस पोस्ट को अपने दोस्तों के साथ share करें. धन्यवाद.

---

## What is Unit Testing in Hindi?

by yugal joshi - Monday, July 18, 2016

<https://ehindistudy.com/2016/07/18/unit-testing-hindi/>

### यूनिट टेस्टिंग क्या है?:-

यूनिट टेस्टिंग सॉफ्टवेर टेस्टिंग की एक विधि है जिसमें सॉफ्टवेर एप्लीकेशन के सबसे छोटे भाग (जिन्हें हम units कहते हैं) को test किया जाता है.

आसान शब्दों में कहें तो, “unit testing एक ऐसी टेस्टिंग है जिसमें प्रोग्राम को टुकड़ों में तोड़ लिया जाता है, तथा प्रत्येक टुकड़े को बारीकी से test किया जाता है.”

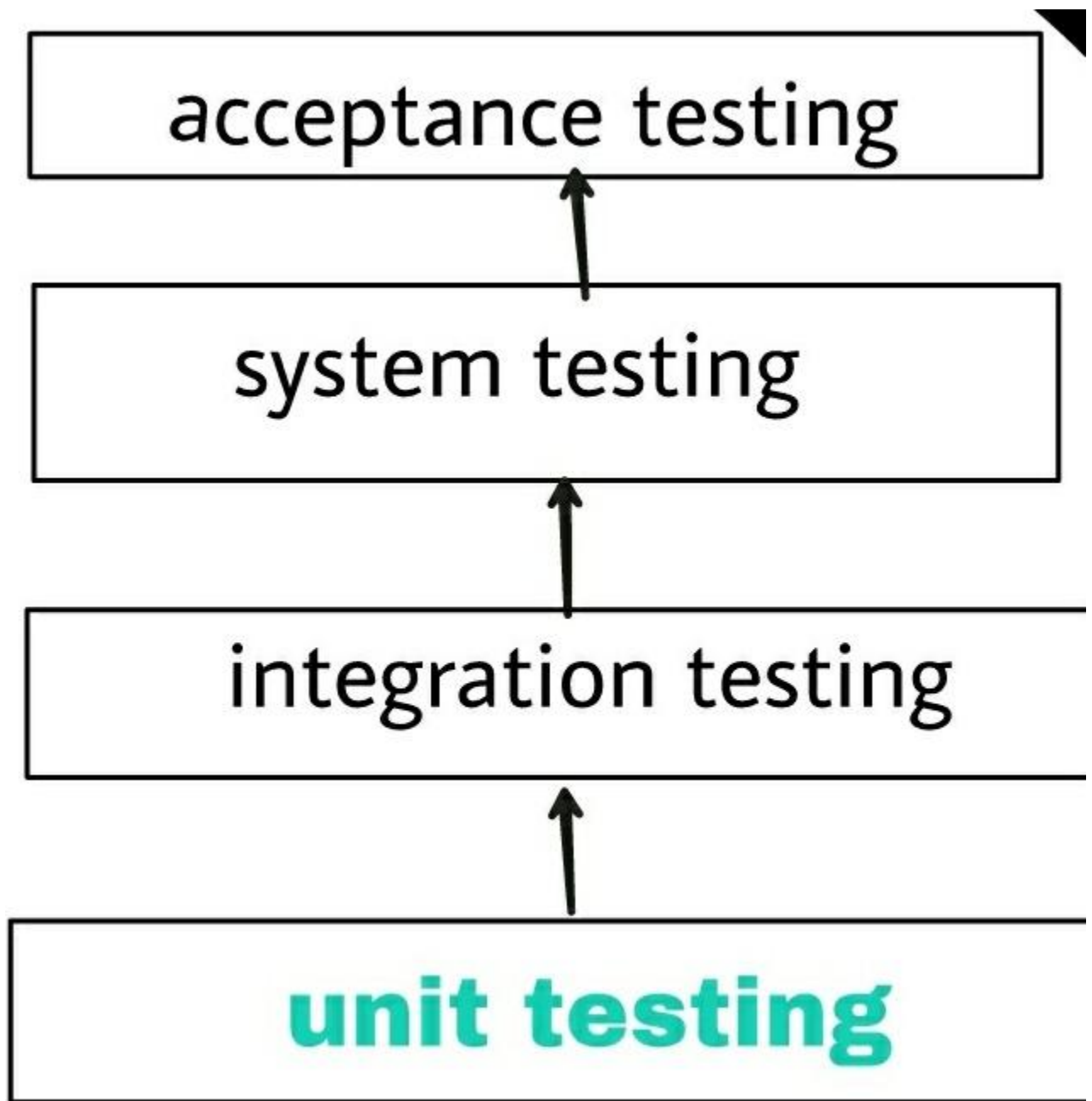
इस टेस्टिंग का मुख्य उद्देश्य यह सुनिश्चित करना है कि सॉफ्टवेर का प्रत्येक यूनिट का source कोड सही है तथा इसे use किया जा सकता है.

पहले adhoc tools का प्रयोग units को test करने के लिए किया जाता था परन्तु आजकल unit test करने के लिए frameworks(जावा फ्रेमवर्क, [.net फ्रेमवर्क](#) तथा PHP फ्रेमवर्क आदि) का प्रयोग किया जाता है.

यूनिट टेस्ट developers के द्वारा लिखे तथा परफॉर्म किये जाते हैं तथा इसे करने के लिए [WHITE BOX TESTING](#) विधि का प्रयोग किया जाता है.

यहाँ याद रखने वाली बात यह है कि “यूनिट टेस्टिंग में बहुत ही अधिक समय लगता है तथा इसमें बहुत ही धैर्य की आवश्यकता होती है.”

यह टेस्टिंग बहुत ही अधिक प्रभावपूर्ण है क्योंकि इसके प्रयोग के द्वारा अधिकतर defects को identify कर लिया जाता है.



सामान्यतया यूनिट टेस्टिंग integration टेस्टिंग के बाद की जाती है.

### **Advantage of unit testing in hindi:-**

यूनिट टेस्टिंग के निम्नलिखित लाभ हैं:-

1:-इस टेस्टिंग के द्वारा हम सॉफ्टवेर में defects तथा bugs को early stages में ही ढूँढ लेते हैं, बाद में defects तथा bugs को ढूँढना बहुत अधिक कठिन हो जाता है.

2:-यह टेस्टिंग कोडिंग की प्रक्रिया को और अधिक effective तथा agile(फुर्तीला) बना देती है जिससे हम सॉफ्टवेर में ज्यादा से ज्यादा features को add कर सकते हैं.

3:-जब हम यूनिट टेस्टिंग कर लेते हैं तो हमें manual टेस्टिंग की आवश्यकता बहुत कम रह जाती है. वैसे भी manual टेस्टिंग बहुत ही अधिक boring तथा खर्चीली है.

4:-हम इस टेस्टिंग के द्वारा सॉफ्टवेर के design को बिना break किये हुए इसके डिज़ाइन को बेहतर बना सकते हैं.

5:-जब हम पहले ही bugs को detect कर लेते हैं तो इससे हमारे समय तथा cost की बचत हो जाती है.

6:-यह टेस्टिंग कोड की efficiency को बढ़ाता है तथा इसको maintain करना आसान हो जाता है.

7:-इस टेस्टिंग का एक फायदा यह है कि यह हमें सॉफ्टवेर के updated documentation(दस्तावेज) उपलब्ध करता है.

---

यह पोस्ट आपको कैसी लगी हमें comment के माध्यम से बताइये तथा अपने friends के साथ share करें। धन्यवाद।

---

## what is V-model in hindi?

by yugal joshi - Thursday, August 02, 2018

<https://ehindistudy.com/2018/08/02/v-model-hindi/>

### V-Model in hindi:-

आज हम इस पोस्ट में v-model के बारे में पढ़ेंगे तो चलिए शुरू करते हैं.

v-model को verification तथा validation (V&V) मॉडल भी कहते हैं.

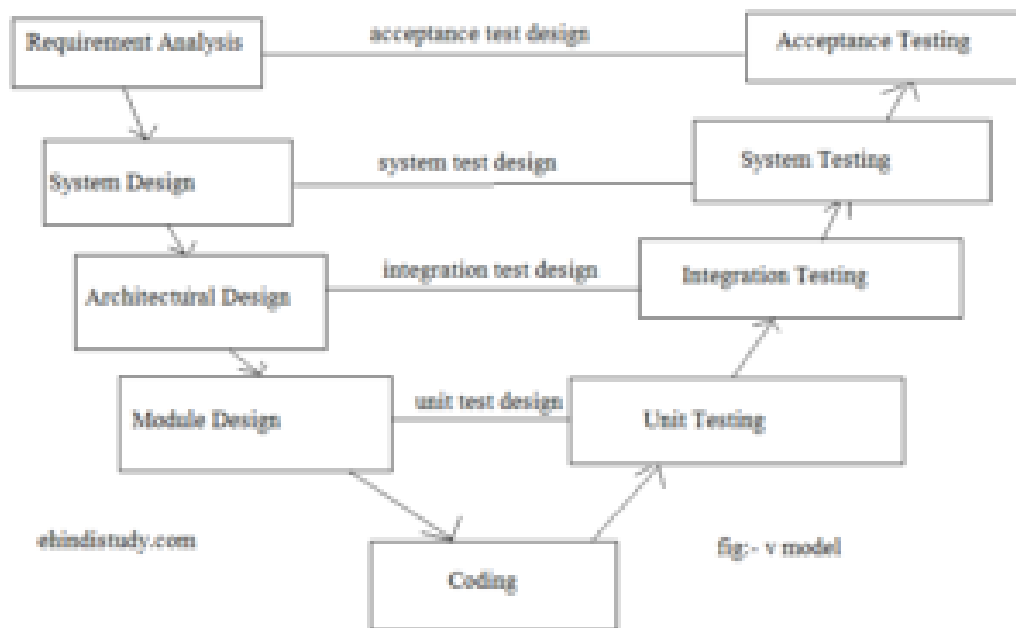
v-model एक [SDLC मॉडल](#) है जहाँ प्रोसेस को V shape में execute किया जाता है.

v-model सॉफ्टवेयर डेवलपमेंट लाइफ साइकिल के प्रत्येक phase के लिए उपयोगी है.

v-model जो है वह [waterfall model](#) का extension है. तथा यह निर्धारित करता है कि प्रत्येक डेवलपमेंट phase टेस्टिंग phase से जुड़ा हुआ है.

इसका अर्थ यह हुआ कि सॉफ्टवेयर डेवलपमेंट के प्रत्येक phase के लिए एक testing phase होता है.

यह बहुत ही ज्यादा disciplined model है इसमें अगला phase तभी शुरू होता है जब तक कि पिछला phase पूरा नहीं हो जाता है.



चित्र में, left side में verification phases हैं तथा right side में validation phases हैं. यह पूरा चित्र V की तरह दिख रहा है इसलिए इसे V-model कहते हैं.

## phases of V-model in hindi

### v-model verification phases:-

**1:- requirements analysis:-** यह डेवलपमेंट साइकिल का पहला phase है इसमें कस्टमर की जरूरतों के हिसाब से प्रोडक्ट की requirements को analyze किया जाता है. इस phase में कस्टमर से अच्छी तरह प्रोडक्ट से सम्बन्धित जरूरतों को इक्कठा किया जाता है.



यह बहुत ही जरूरी फेज है क्योंकि यही फेज आने वाले phases को निर्धारित करता है.

इस फेज में बाद में प्रयोग के लिए acceptance tests को डिजाइन किया जाता है.

**2:- system design:-** जब हमारे पास प्रोडक्ट की requirements आ जाती है तो अब हम सिस्टम का पूरा एक डिजाइन तैयार करते हैं. इसमें प्रोडक्ट बनाने के लिए जरूरी हार्डवेयर तथा सभी technical components का पूरा एक विवरण बना लिया जाता है.

**3:- architectural design:-** इस फेज में architectural specifications को डिजाइन किया जाता है.

इसमें यह specification (विवरण) होता है कि किस प्रकार सॉफ्टवेयर सभी components के साथ internally तथा externally लिंक होगा. इसलिए इस फेज को high level design (HLD) भी कहते हैं.

**4:- module design:-** इस फेज में सिस्टम के सभी modules के internal design को specify किया जाता है. इसलिए इसे low level design (LLD) कहते हैं.

यह बहुत जरूरी होता है कि सभी modules का डिजाइन सिस्टम आर्किटेक्चर के मुताबिक हो.

module design फेज में unit tests को भी डिजाइन कर लिया जाता है.

**5:- coding phase:-** coding phase में, पिछले phases में किये गये डिजाइन तथा specification की कोडिंग की जाती है. इस फेज में सबसे अधिक समय लगता है.

कोडिंग के लिए उपयुक्त प्रोग्रामिंग लैंग्वेज का चुनाव किया जाता है तथा code को लिखते वक्त उनका बार बार review किया जाता है.

### **validation phases:-**

**1:- unit testing:-** यूनिट टेस्टिंग फेज में, module design फेज के दौरान बनाये गये unit tests को execute किया जाता है. unit testing कोड लेवल की टेस्टिंग है यह सिर्फ technical design को ही verify करती है. इसलिए यह सभी defects को टेस्ट नहीं कर पाती है.

**2:- integration testing:-** [integration टेस्टिंग](#) में, architectural design फेज में बनाये गये integration tests को execute किया जाता है.

integration testing यह सुनिश्चित करती है कि सभी modules आपस में अच्छी तरह कार्य कर रहे हैं.

**3:- system testing:-** सिस्टम टेस्टिंग में, system design फेज में बनाये गये system tests को execute किया जाता है. system tests सिस्टम की पूरी functionality को check करता है.

इस में ज्यादा performance testing तथा regression testing पर ध्यान दिया जाता है.

**4:- acceptance testing:-** acceptance टेस्टिंग में, requirement analysis फेज में बनाये गये acceptance tests को execute किया जाता है.

यह testing यह सुनिश्चित करती है कि सिस्टम दूसरे सिस्टम के साथ compatible है. तथा इसमें non-functional issues जैसे:- load time, performance आदि को user environment में टेस्ट किया जाता है.

## **Advantage of v-model in hindi:-**

v-model के लाभ निम्नलिखित हैं:-

1:- यह बहुत ही ज्यादा disciplined model है इसमें एक समय में केवल एक phase को ही पूरा किया जाता है.

2:- यह बहुत ही सरल model है इसे आसानी से समझा तथा प्रयोग किया जा सकता है.

3:- इस model की stiffness (कठोरता) के कारण इसे आसानी से manage किया जा सकता है.

4:- इसमें planning तथा tests डिजाइनिंग, कोडिंग से पहले हो जाती है जिससे समय बचता है.

5:- इसमें defects का early stages में ही पता चल जाता है.

6:- यह छोटे projects के लिए बहुत अच्छा है जहाँ requirements को अच्छी तरह समझा जा सकता है.

7:- इसमें resources को अच्छी तरह utilize किया जा सकता है.

### **disadvantage of v-model in hindi:-**

v-model के disadvantage निम्नलिखित हैं.

1:- इसमें बहुत ही ज्यादा risk और अनिश्चितता रहती है.

2:- बड़े, कठिन तथा ऑब्जेक्ट ओरिएंटेड projects के लिए यह model सही नहीं है.

3:- यह flexible model नहीं है.

4:- इसमें अगर बीच में कोई बदलाव कर दिया तो test डोक्यूमेंट्स को भी change करना पड़ता है.

**निवेदन:-** आपको यह पोस्ट पसंद आई हो तो इसे अपने दोस्तों के साथ अवश्य share करें.  
धन्यवाद.

---

## **What is walkthrough in Hindi?**

by yugal joshi - Monday, July 04, 2016

<https://ehindistudy.com/2016/07/04/what-is-walkthrough-in-hindi/>

### **Walkthrough in software testing in hindi:-**

>

यह एक अनौपचारिक review है तथा इसमें किसी भी तैयारी की आवश्यकता नहीं होती है।

इसमें एक डिज़ाइनर या प्रोग्रामर डेवलपमेंट टीम को lead करता है और जो इसमें प्रतिभागी(participants) होते हैं वह सॉफ्टवेयर में आने वाली संभावित त्रुटियों, सॉफ्टवेयर standards का violation तथा अन्य परेशानियों के बारे में सवाल पूछते हैं तथा उन परेशानियों के लिए सुझाव भी देते हैं।

Walkthrough सामान्यतया author के द्वारा organise किया जाता है। इसमें author सॉफ्टवेयर प्रोडक्ट कोड को पढ़ता है तथा अन्य टीम के सदस्य उसमें उपस्थित त्रुटियों तथा सुझावों को बताते हैं।

walkthrough  
kya hai?

#### Walkthrough का उद्देश्य:-

इसको निम्नलिखित उद्देश्य के लिए conduct किया जाता है:-

- 1:-इसका प्रयोग त्रुटियों को ढूँढने में किया जाता है।
- 2:-परेशानियों के वैकल्पिक समाधान को discuss करने में इसका प्रयोग किया जाता है।
- 3:-इसमें इस बात पर ध्यान दिया जाता है कि कैसे सॉफ्टवेयर की जरूरतों को पूरा किया जायें।

walkthrough में मुख्य तीन विशेषज्ञ होते हैं जो निम्नलिखित हैं:-

**1:-Author:-**walkthrough मीटिंग में author सॉफ्टवेयर प्रोडक्ट को प्रस्तुत करता है और काम को पूरा करने की जिम्मेदारी author ही handle करता है।

**2:-Leader:-**इसको लीडर के द्वारा आयोजित किया जाता है administration सभी कार्य लीडर ही करता है।

**3:-Recorder:-**Walkthrough मीटिंग में चिन्हित किये जाने वाली सभी bugs, defects, तथा उसमें लिए जाने वाले निर्णय, actions को रिकॉर्ड करने काम रिकॉर्डर करता है।

अगर यह पोस्ट आपके लिए helpful रही है तो आप अपने friends के साथ share कर सकते हैं। आपके कोई suggestions या questions हैं तो आप comment के माध्यम से बताएँ।

---

**ehindistudy.com**

**computer science notes in hindi**