



IT - ITeS SSC  
NASSCOM

# प्रतिभागी पुस्तिका

क्षेत्र  
आईटी-आईटीईएस

उप-क्षेत्र  
सॉफ्टवेयर उत्पाद (प्रोडक्ट)

व्यवसाय  
उत्पाद विकास और वितरण

Reference ID: SSC/Q6702, Version 3.0  
NSQF level: 5



ईबुक तक पहुंचने के लिए इस क्यूआर कोड को  
स्कैन/क्लिक करें

## सॉफ्टवेयर डेवलपर - प्रोडक्ट डेवलपमेंट

## प्रकाशक

### आईटी-आईटीईएस सेक्टर स्किल कौंसिल नैसकॉम

पता: प्लाट नंबर- 7, 8, 9, & 10, सेक्टर- 126, नोएडा, उत्तर प्रदेश - 201303

नई दिल्ली - 110049

वेबसाइट: [www.sscnasscom.com](http://www.sscnasscom.com)

ईमेल: [ssc@nasscom.com](mailto:ssc@nasscom.com)

फोन नं: 0120 4990111 - 0120 4990172

पहला संस्करण, दिसंबर 2022

कॉपीराइट © 2022

यह पुस्तक आईटी-आईटीईएस सेक्टर स्किल कौंसिल नैसकॉम द्वारा प्रायोजित है।

क्रिएटिव कॉमन्स लाइसेंस के तहत: CC-BY-SA

### Attribution-ShareAlike: CC BY-SA



यह लाइसेंस अन्य लोगों को व्यावसायिक उद्देश्यों के लिए भी आपके काम को रीमिक्स, ट्वीक और निर्माण करने देता है, जैसे जब तक वे आपको श्रेय देते हैं और समान शर्तों के तहत अपनी नई रचनाओं का लाइसेंस देते हैं। यह लाइसेंस अक्सर फ्लॉपीलेफ्ट मुक्त और ओपन सोर्स सॉफ्टवेयर लाइसेंस से तुलनात्मक है। इस पर आधारित सभी नए कार्य पर समान लाइसेंस होगा, इसलिए कोई भी डेरिवेटिव व्यावसायिक उपयोग की भी अनुमति देगा। यह विकिपीडिया द्वारा उपयोग किया जाने वाला लाइसेंस है और उन सामग्रियों के लिए अनुशंसित जो विकिपीडिया और इसी तरह के लाइसेंस प्राप्त परियोजनाओं से सामग्री शामिल करने से लाभान्वित होगा।

## अस्वीकरण

इसमें निहित जानकारी विभिन्न विश्वसनीय स्रोतों से प्राप्त की गई है। आईटी-आईटीईएस सेक्टर स्किल कौंसिल नैसकॉम ऐसी जानकारी की सटीकता, पूर्णता या पर्याप्तता के लिए सभी वारंटी को अस्वीकार करता है। नैसकॉम की यहां निहित जानकारी में त्रुटियों, चूक या अपर्याप्तता के लिए, या उसकी व्याख्या के लिए कोई दायित्व नहीं होगा। पुस्तक में शामिल कॉपीराइट सामग्री के स्वामी का पता लगाने का हर संभव प्रयास किया गया है। पुस्तक के भविष्य के संस्करणों में पावती के लिए उनके ध्यान में लाए गए किसी भी चूक के लिए प्रकाशक आभारी होंगे। नैसकॉम की कोई भी संस्था इस सामग्री पर निर्भर रहने वाले किसी भी व्यक्ति को हुए किसी भी प्रकार के नुकसान के लिए जिम्मेदार नहीं होगी। दिखाये गए सभी चित्र केवल द्रष्टांत उद्देश्य के लिए हैं। क्विक रिस्पॉन्स कोड (क्यूआर कोड) पुस्तक में कोडित बॉक्स सामग्री से जुड़े ई संसाधनों तक पहुंचने में मदद करेंगे। ये क्यूआर कोड विषय में ज्ञान बढ़ाने के लिए इंटरनेट पे उपलब्ध लिंक और यूट्यूब वीडियो संसाधनों से उत्पन्न होते हैं और नैसकॉम द्वारा नहीं बनाए गए हैं। सामग्री में लिंक या क्यूआर कोड को एम्बेड करना किसी भी प्रकार का समर्थन नहीं माना जाना चाहिए। व्यक्त किए गए विचारों या लिंक किए गए वीडियो की सामग्री या विश्वसनीयता के लिए नैसकॉम जिम्मेदार नहीं है। नैसकॉम गारंटी नहीं दे सकता कि ये लिंक/क्यूआर कोड हर समय काम करेंगे क्योंकि लिंक किए गए पृष्ठों की उपलब्धता पर हमारा कोई नियंत्रण नहीं है।





श्री नरेंद्र मोदी  
भारत के प्रधानमंत्री

“

कौशल विकास भारत को एक बेहतर देश  
बना रहा है। अगर हमें भारत को  
विकसित करना है तो  
कौशल विकास हमारा लक्ष्य होना चाहिए।

”



## Certificate

### COMPLIANCE TO QUALIFICATION PACK – NATIONAL OCCUPATIONAL STANDARDS

is hereby issued by the

**IT – ITeS Sector Skill Council NASSCOM**

for

### SKILLING CONTENT: PARTICIPANT HANDBOOK

Complying to National Occupational Standards of  
Job Role/ Qualification Pack: '**Software Developer-Product Development**'  
QP No. '**SSC/Q6702, NSQF Level 5**'

Date of Issuance: **September 13<sup>th</sup>, 2021**

Valid up to: **September 13<sup>th</sup>, 2024**

\* Valid up to the next review date of the Qualification Pack

Authorised Signatory  
(IT – ITeS Sector Skill Council NASSCOM)

## आभार-पूर्ति

आईटी-आईटीईएस क्षेत्र कौशल परिषद नैसकॉम उन सभी व्यक्तियों और संस्थानों के प्रति अपनी कृतज्ञता व्यक्त करना चाहता है जिन्होंने इस "पारंपरिक हैंडबुक" की तैयारी के लिए विभिन्न तरीकों से योगदान दिया है। इसमें उनके योगदान के बिना यह पूरा नहीं होता। उन लोगों को विशेष धन्यवाद दिया जाता है जिन्होंने इसके विभिन्न मॉड्यूल की तैयारी में सहयोग किया। उन सभी की भी ईमानदारी से सराहना की जाती है जिन्होंने इन मॉड्यूलों के लिए सहकर्मी समीक्षा प्रदान की। इस हैंडबुक को तैयार करना आईटी-आईटीईएस समर्थन के बिना संभव नहीं होता। शुरुआत से अंत तक उद्योग की प्रतिक्रिया बेहद उत्साहजनक रही है और यह उनके इनपुट के साथ है कि हमने उद्योग में आज मौजूद कौशल अंतराल को पाटने की कोशिश की है।

यह पुस्तिका उन महत्वाकांक्षी युवाओं को समर्पित है जो विशेष कौशल हासिल करना चाहते हैं जो उनके भविष्य के प्रयासों के लिए आजीवन संपत्ति होगी।

## इस पुस्तक के बारे में

यह प्रतिभागी हैंडबुक एक सोवेयर डेवलपर के रूप में काम करने के लिए आवश्यक कौशल और ज्ञान के लिए प्रशिक्षण प्रदान करने के लिए विकसित की गई है। इसे राष्ट्रीय कौशल योग्यता ढांचे के तहत योग्यता पैक SSC/Q6702 के आधार पर डिज़ाइन किया गया है और इसमें निम्नलिखित NOS/विषय शामिल हैं:

1. SSC/N0502 - विशिष्टता के लिए इतना वेयर कोड विकसित करें
2. SSC/N9014 - समावेशी और पर्यावरणीय रूप से सतत कार्यस्थल रोज़गार कौशल
3. DGT/VSQ/N0102 - रोज़गार कौशल

यह पुस्तक प्रतिभागियों की तकनीकी मांग और शैक्षिक पृष्ठभूमि को ध्यान में रखते हुए तैयार की गई है। इस प्रकार, ज्यादातर ग्राफिक्स और चित्रण के माध्यम से नौकरी के लिए आवश्यक अवधारणाओं को समझाने के लिए विशेष प्रयास किया गया है।

## इस्तेमाल किया गया प्रतीक



मुख्य सीख



यूनिट के  
उद्देश्य



अभ्यास



नोट्स



गतिविधि



सारांश

## विषय - सूची

क्रम	विषय और यूनिट	पृष्ठ संख्या
1.	<b>सॉफ्टवेयर उत्पाद उद्योग- परिचय</b>	1
	यूनिट 1.1 - आईटी-आईटीईएस उद्योग और इसके विभिन्न उप-क्षेत्रों का परिचय	3
	यूनिट 1.2 - एक सॉफ्टवेयर डेवलपर के लिए करियर पथ	13
2.	<b>सॉफ्टवेयर डेवलपमेंट का परिचय</b>	23
	यूनिट 2.1 - सॉफ्टवेयर विकास और आईटी सेवाएं	25
3.	<b>सॉफ्टवेयर डेवलपमेंट की प्रक्रिया (SSC/N0502)</b>	37
	यूनिट 3.1 - सॉफ्टवेयर डेवलपमेंट प्रक्रिया	39
4.	<b>अनुप्रयोग (एप्लीकेशन) प्रबंधन की प्रक्रिया (SSC/N0502)</b>	49
	यूनिट 4.1 - सॉफ्टवेयर आवश्यकताएँ विशिष्टता की गुणवत्ता विशेषताएँ	51
	यूनिट 4.2 - कस्टम और रैपिड एप्लिकेशन सॉफ्टवेयर	84
5.	<b>पुनः - प्रयोज्य कोड डेवलपमेंट की अवधारणा (SSC/N0502)</b>	103
	यूनिट 5.1 - सॉफ्टवेयर परीक्षण	105
6.	<b>प्रोग्रामिंग और सॉफ्टवेयर डेवलपमेंट के लिए एल्गोरिथम (SSC/N0502)</b>	123
	यूनिट 6.1 - सॉफ्टवेयर प्रोग्रामिंग और एल्गोरिथम डेवलपमेंट	125
7.	<b>सॉफ्टवेयर अनुप्रयोगों की डिजाइन (SSC/N0502)</b>	143
	यूनिट 7.1 - सॉफ्टवेयर डेवलपमेंट लाइफ साइकल (चक्र)	145
	यूनिट 7.2 - सॉफ्टवेयर डेवलपमेंट प्रक्रिया को मापने के लिए एलिमेंट	159
8.	<b>नौकरी की भूमिका के लिए कार्य आवश्यकताएँ, उपकरण और सॉफ्टवेयर (SSC/N0502)</b>	165
	यूनिट 8.1 - कार्यक्रम विशिष्टता और प्रोग्रामिंग भाषा	167
	यूनिट 8.2 - अनुप्रयोगों और प्रमुख प्रक्रियाओं को विकसित करने के लिए दृष्टिकोण	180



क्रम	विषय और यूनिट	पृष्ठ संख्या
9.	समावेशी और पर्यावरण की दृष्टि से टिकाऊ कार्यस्थल (SSC/N9014)	193
	यूनिट 9.1 - सतत अभ्यास	195
	यूनिट 9.2 - विविधता का सम्मान करें और समानता को बढ़ावा देने के लिए प्रथाओं को मजबूत करें	207
10.	रोजगार कौशल (DGT/VSQ/N0102)	219
11.	अनुलग्नक	223





IT - ITeS SSC  
NASSCOM

# 1. सॉफ्टवेयर उत्पाद उद्योग- परिचय

- यूनिट 1.1 आईटी-आईटीईएस उद्योग और इसके विभिन्न उप-क्षेत्रों का परिचय  
यूनिट 1.2 एक सॉफ्टवेयर डेवलपर के लिए करियर पथ



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. आईटी/सॉफ्टवेयर उत्पाद विकास उद्योग में उपयोग किए जाने वाले विभिन्न वितरण मॉडलों को समझने में
2. प्रमुख अनुप्रयोगों (एप्लीकेशन) को जहां सॉफ्टवेयर विकास का उपयोग किया जाता है उन्हें वर्गीकृत करने में

## यूनिट 1.1: आईटी-आईटीईएस उद्योग और इसके विभिन्न उप-क्षेत्रों का परिचय

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. आईटी-आईटीईएस क्षेत्र की प्रासंगिकता पर चर्चा करें

### 1.1.1 आईटी-आईटीईएस उद्योग का परिचय

लगभग हर प्रमुख वैश्विक उद्योग सूचना प्रौद्योगिकी और आईटी उद्योग से जुड़े हार्डवेयर और सॉफ्टवेयर पर निर्भर करते हैं। सूचना प्रौद्योगिकी उद्योग दुनिया में सबसे समृद्ध सेक्टर में से एक है जो लगातार विकसित हो रहा है। आईटी ने किसी भी अन्य उद्योग या आर्थिक क्षेत्र की तुलना में उत्पादकता में वृद्धि की है, विशेष रूप से विकसित दुनिया में, और इस प्रकार वैश्विक आर्थिक विकास का एक प्रमुख चालक है। यह तेजी से विस्तार करने वाला उद्योग पैमाने की अर्थव्यवस्थाओं और अतृप्त उपभोक्ता और व्यावसायिक मांग द्वारा प्रतिष्ठित है। सूचना प्रौद्योगिकी एसोसिएशन ऑफ अमेरिका (आईटीएए) द्वारा "सूचना प्रौद्योगिकी" को "कंप्यूटर आधारित सूचना प्रणाली के सभी संभावित पहलुओं" के रूप में परिभाषित किया गया है। आईटी उद्योग में कंप्यूटर सिस्टम से लेकर डिजाइन, कार्यान्वयन, अनुसंधान, सॉफ्टवेयर और हार्डवेयर विकास तक सब कुछ शामिल है।



चित्र 1.1 आईटी उद्योग की विशेषताएं

चूंकि यह सूचना तक आसान पहुंच प्रदान करता है, आईटी उद्योग ई-गवर्नेंस के लिए एक माध्यम के रूप में काम कर सकता है। सेवा क्षेत्र में सूचना प्रौद्योगिकी के अनुप्रयोग से परिचालन दक्षता और पारदर्शिता बढ़ती है। यह कौशल विकास के माध्यम के रूप में भी कार्य करता है।

## 1.1.2 आईटीईएस उद्योग और इसके विभिन्न उप-क्षेत्र

आईटी उद्योग ने भारतीय शिक्षा क्षेत्र में विशेष रूप से इंजीनियरिंग और कंप्यूटर विज्ञान में मांग को बढ़ावा दिया है। भारत में आईटी और आईटीईएस उद्योग चार प्रमुख खंडों में विभाजित है: आईटी सेवाएं, व्यवसाय प्रक्रिया प्रबंधन (बीपीएम), सॉफ्टवेयर उत्पाद और इंजीनियरिंग सेवाएं और हार्डवेयर को शामिल किया गया है।

निम्नलिखित आंकड़ा आईटी / आईटीईएस क्षेत्र के उप-क्षेत्रों के रूप में पहचाने जाने वाले मुख्य उप-क्षेत्रों को प्रदर्शित करता है:



### आईटी सेवा

• इसमें विभिन्न जुड़ाव श्रेणियां शामिल हैं, जैसे परामर्श, सिस्टम एकीकरण, आईटी आउटसोर्सिंग/प्रबंधित सेवाएं/होस्टिंग सेवाएं, प्रशिक्षण, और समर्थन/रखरखाव शामिल है।

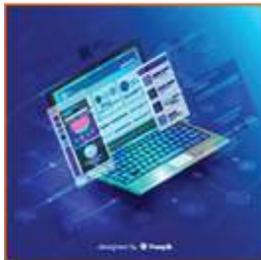


### विजनेस प्रक्रिया प्रबंधन

यह परिभाषित और मापने योग्य प्रदर्शन मेट्रिक्स के तहत चयनित प्रक्रियाओं का स्वामित्व और प्रबंधन करता है।

इस उप-क्षेत्र का विकास उच्च-लागत से कम-लागत वाले गंतव्यों तक व्यावसायिक प्रक्रियाओं के वितरण में बदलाव का प्रतिनिधित्व करता है।

यह बदलाव सूचना और संचार प्रौद्योगिकियों में प्रगति से सुगम हुआ है।



### सॉफ्टवेयर उत्पाद

सॉफ्टवेयर उत्पाद किसी भी प्रकार का प्रोग्राम या कोड सेट है जो व्यावसायिक रूप से खरीद, पट्टे, किराये या सेवा के लिए उपलब्ध है।

पैकेज्ड सॉफ्टवेयर राजस्व में आम तौर पर प्रारंभिक और चल रहे राइट-टू-यूज पैकेज्ड सॉफ्टवेयर लाइसेंस के लिए शुल्क शामिल होते हैं।



### इंजीनियरिंग अनुसंधान एवं विकास

यह किसी उत्पाद या सेवा के निर्माण, जीवन काल को अधिकतम करने और उत्पाद / संपत्ति की उपज को अनुकूलित करने से जुड़ा है।

इसमें न केवल उत्पाद या सेवा के डिजाइन तत्व शामिल हैं बल्कि विनिर्माण / वितरण में शामिल बुनियादी ढांचे, उपकरण और प्रक्रियाएं भी शामिल हैं।

•

चित्र 1.2 आईटी/आईटीईएस उद्योग के उप-क्षेत्र



चित्र 1.3 ITeSसेक्टर

हाल के एक आर्थिक सर्वेक्षण के अनुसार, आईटी-आईटीईएस क्षेत्र ने पिछले चार वर्षों में प्रति वर्ष लगभग चौदह मिलियन नौकरियां प्रदान की हैं, और यह आंकड़ा भविष्य में बढ़ने की उम्मीद है। तकनीकी जनशक्ति की मांग और आपूर्ति के बीच का अंतर इस तरह की अभूतपूर्व वृद्धि दर के प्राथमिक कारणों में से एक है। इस क्षेत्र की उच्च नौकरी छोड़ने की दर के परिणामस्वरूप अक्सर बड़ी संख्या में रोजगार के अवसर पैदा होते हैं और भारत में विभिन्न क्षेत्रों में बहुत सारे अवसर मिलते हैं।

### 1.1.3 आईटी/सॉफ्टवेयर उत्पाद विकास में प्रयुक्त विभिन्न वितरण मॉडल

सुविधाओं और अतिरिक्त मूल्य के साथ एक नया उत्पाद बनाना सॉफ्टवेयर उत्पाद विकास कहलाता है। इसमें एक पूरी तरह से नया उत्पाद बनाना, मौजूदा को परिष्कृत करना, या यहां तक कि विकास के लिए सिस्टम के तरीकों को बदलना भी शामिल हो सकता है।

नए बनाए गए सॉफ्टवेयर आइटम के साथ, बाजार बदल सकता है और उपयोगकर्ताओं को एक बेहतर समग्र अनुभव होगा। ग्राहकों को लाभ होगा क्योंकि विकास और नवाचार नए उत्पाद नवाचारों का मार्ग प्रशस्त करते हैं। सॉफ्टवेयर उत्पाद विकास एक कठोर, निरंतर प्रक्रिया है जो व्यक्तियों या व्यवसायों के लिए एक विशिष्ट विधि, उद्देश्य या उद्देश्य का प्रतिनिधित्व करना चाहता है। एक कार्यात्मक सॉफ्टवेयर उत्पाद के निर्माण के लिए अक्सर एक संपूर्ण कार्यप्रणाली की आवश्यकता होती है जिसमें कई सॉफ्टवेयर विकास चरण या विधियां शामिल होती हैं।

सॉफ्टवेयर डिलीवरी मॉडल वे रणनीतियाँ हैं जिनका उपयोग टीम सॉफ्टवेयर उत्पाद को बाजार के लिए तैयार करने के लिए करती है। सॉफ्टवेयर वितरण जीवनचक्र, सॉफ्टवेयर वितरण पाइपलाइन, या केवल सॉफ्टवेयर वितरण प्रक्रिया इन दृष्टिकोणों के अन्य नाम हैं।

आवश्यक लक्ष्यों को पूरा करने के लिए विभिन्न विकास जीवन चक्र मॉडल बनाए जाते हैं। निष्पादन के क्रम के साथ-साथ प्रक्रिया और उसके चरणों को मॉडल में वर्णित किया गया है। मॉडल की पसंद का परीक्षण पर महत्वपूर्ण प्रभाव पड़ता है। यह परिभाषित करेगा कि क्या, कहाँ, और कब इच्छित परीक्षण प्रतिगमन परीक्षण को प्रभावित करेगा और बड़े पैमाने पर यह निर्धारित करेगा कि कौन सी परीक्षण पद्धति को लागू करना है। सॉफ्टवेयर विकास में उपयोग किए जाने वाले कई मॉडल या दृष्टिकोण हैं। वे इस प्रकार हैं:



चित्र 1.4 आईटी/सॉफ्टवेयर उत्पाद विकास के विभिन्न डिलीवर मॉडल

### 1.1.4 व्यवसाय प्रक्रिया प्रबंधन (बीपीएम)

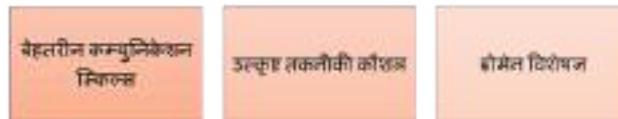
बीपीएम उद्योग भारत के सॉफ्टवेयर और सेवा उद्योगों के विकास के प्रमुख चालक के रूप में उभरा है। ग्राहक सेवा, तकनीकी सहायता, टेलीमार्केटिंग, ऑनलाइन शोध आदि जैसी सेवाओं की पेशकश करते हुए, आईटीईएस-बीपीएम क्षेत्र भारतीय उद्योग में सबसे बड़े रोजगार सृजनकर्ताओं में से एक के रूप में उभरा है। परीक्षण या गुणवत्ता आश्वासन, अनुप्रयोग विकास और रखरखाव, और डेटा सेंटर संचालन सभी एक तीसरे पक्ष के विक्रेता / कंपनी को आउटसोर्स किए जाते हैं ताकि परिचालन लागत कम हो और दक्षता बढ़े। कुछ प्रमुख बीपीएम आउटसोर्सिंग सेवा प्रदाताओं में जेनपैक्ट, विप्रो, टीसीएस और इंफोसिस शामिल हैं।



चित्र 1.5 व्यवसाय प्रक्रिया प्रबंधन (बीपीएम) के लिए आवश्यक कौशल

इंजीनियरिंग सर्विसेज आउटसोर्सिंग (ईएसओ): यह सबसे बड़े औद्योगिक क्षेत्रों में से एक है और इसके 2020 तक \$40 बिलियन तक बढ़ने की उम्मीद है। इंजीनियरिंग सेवाओं को आमतौर पर दो कारणों से आउटसोर्स किया जाता है: लागत और गुणवत्ता। बीपीएम क्षेत्र के विपरीत, जहां आउटसोर्सिंग सेवाओं में कम परिचालन लागत एक प्रमुख कारक है, ईएसओ एक डोमेन-केंद्रित कंपनी या तीसरे पक्ष से उच्च गुणवत्ता वाले उत्पाद प्राप्त करने की क्षमता पर केंद्रित है।

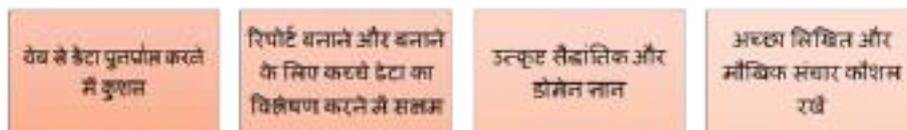
इंजीनियरिंग सेवाओं की डिलीवरी आवश्यकता चरण (नए उत्पाद या प्रौद्योगिकी के निर्माण के लिए) से शुरू होती है। डिजाइन चरण आवश्यकता चरण का अनुसरण करता है, जिसमें इंजीनियर डिजाइन विनिर्देशों और दिशानिर्देशों के अनुसार विचार या समस्या और डिजाइन समाधान का विश्लेषण करते हैं। डिजाइन किया गया मॉडल परीक्षण से पहले निर्माण या निर्माण चरण से गुजरता है। इंजीनियरिंग के पूरे चक्र में मानक महत्वपूर्ण हैं, जैसा कि इंजीनियरिंग परिणाम हैं, और इंजीनियर सभी चरणों में क्लाइंट के साथ संवाद करते हैं। उत्पाद के पूरा होने के बाद इंजीनियर अक्सर उसके संचालन और समर्थन सेवाओं में शामिल होते हैं।



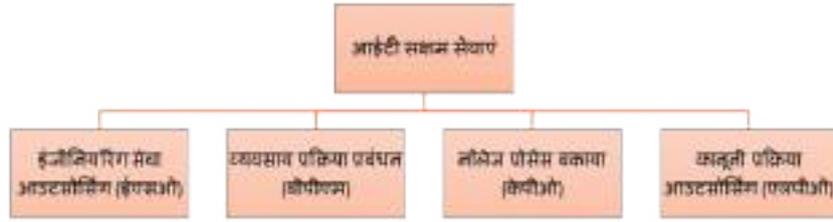
चित्र 1.6 इंजीनियरिंग सेवा आउटसोर्सिंग (ईएसओ) के लिए आवश्यक कौशल

ज्ञान प्रक्रिया आउटसोर्सिंग (केपीओ): इसमें बीपीएम, अनुसंधान और विश्लेषण के तत्व शामिल हैं। ज्ञान प्रसंस्करण में शामिल एक कंपनी इस प्रकार एक ऐसा संगठन होगा जिसका लक्ष्य डोमेन ज्ञान, कौशल और प्रक्रियाओं को करने वाले लोगों के अनुभव से निर्धारित होता है। उदाहरण के लिए, एक अमेरिकी कंपनी बीपीएम पहल के लिए डेटा प्रविष्टि को आउटसोर्स कर सकती है और मानदंड/मानकों के एक सेट के आधार पर अपने अनुप्रयोगों का मूल्यांकन करने के लिए एक केपीओ सेवा प्रदाता का चयन कर सकती है।

अनुप्रयोगों के मूल्यांकन और परीक्षण के लिए काम करने के लिए डेटा प्रविष्टि की तुलना में अधिक कुशल श्रमिकों की आवश्यकता होगी। नतीजतन, केपीओ उद्योग में बीपीओ उद्योग की तुलना में अधिक विशेषज्ञता शामिल है, जो आकार और दक्षता से अधिक चिंतित है।

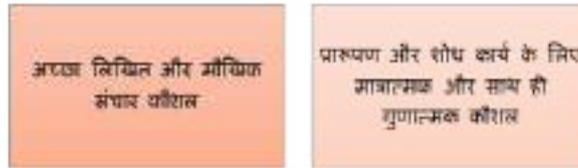


चित्र 1.7 ज्ञान प्रक्रिया आउटसोर्सिंग (केपीओ) के लिए आवश्यक कौशल



चित्र 1.8 IT सक्षम सेवाएं

कानूनी प्रक्रिया आउटसोर्सिंग (एलपीओ): यह कानूनी प्रक्रिया के विभिन्न तत्वों को भारत में अपतटीय केंद्रों को आउटसोर्सिंग / ऑफशोरिंग करने वाली कानूनी फर्मों को संदर्भित करता है। 1995 में, बिकेल एंड ब्रेवर भारत में कानूनी आउटसोर्सिंग करने वाली पहली फर्म थी। छह साल बाद, 2001 में, GE अपने इन-हाउस कानूनी कार्य को संयुक्त राज्य को आउटसोर्स करने वाली पहली कंपनी बन गई। दस्तावेज़ीकरण और प्रारूपण (डेटा प्रविष्टि और दस्तावेज़ प्रबंधन), पेटेंट आवेदन दाखिल करना, उल्लंघन अध्ययन, और कानूनी शोध एक एलपीओ फर्म में सभी सामान्य कार्य हैं। अधिकांश काम के लिए कानून की डिग्री की आवश्यकता होती है, लेकिन इंजीनियर अब एलपीओ क्षेत्र में काम का एक बड़ा हिस्सा करते हैं। पेटेंट कार्य करने वाले इंजीनियरों की बौद्धिक संपदा अधिकार क्षेत्र में अत्यधिक मांग है।



चित्र 1.9 कानूनी प्रक्रिया आउटसोर्सिंग (एलपीओ) के लिए आवश्यक कौशल

### 1.1.5 आईटी-आईटीईएस उद्योग में रुझान

सूचना प्रौद्योगिकी उद्योग पहले की तरह फलफूल रहा है, और अधिक से अधिक ब्रांड इस क्षेत्र में संभावनाओं की भारी मात्रा के कारण विस्तार करना चाह रहे हैं। सूचना प्रौद्योगिकी में अनुप्रयोगों की एक विस्तृत श्रृंखला है, यही वजह है कि यह समय संरचना में एक महत्वपूर्ण महत्वपूर्ण घटक साबित हुई है जो अब उद्योगों के पास है। सूचना प्रौद्योगिकी के बढ़ते महत्व के साथ, उन प्रमुख घटकों को समझना महत्वपूर्ण है जो उपकरणों को क्रांतिकारी बनाते हैं। निम्नलिखित आंकड़ा आईटी और सॉफ्टवेयर विकास उद्योग के प्रमुख रुझानों की व्याख्या करता है।



#### क्लाउड कम्प्यूटिंग

- क्लाउड कम्प्यूटिंग एक डिजिटल वातावरण में अपने संचालन को बेहतर बनाने और सुव्यवस्थित करने के लिए देख रहे ब्रांडों के लिए जाने-माने समाधान है। लाउड कम्प्यूटिंग इंटरनेट के माध्यम से सर्वर, स्टोरेज, डेटाबेस, नेटवर्किंग, सॉफ्टवेयर, एनालिटिक्स और इंटेलिजेंस जैसी कम्प्यूटिंग सेवाओं की डिलीवरी है। "क्लाउड" तेजी से नवाचार, अधिक लचीले संसाधन और पैमाने की अर्थव्यवस्थाएं प्रदान करने के लिए।



#### मोबाइल क्षुधा

- दुनिया भर में ब्रांड और उद्योग मोबाइल ऐप का उपयोग करके और नए संसाधनों को लागू करके अपने काम को बेहतर बनाने के तरीके खोजने का प्रयास कर रहे हैं जो चलते-फिरते काम को और अधिक कुशल बना सकते हैं।



#### बिग डेटा एनालिटिक्स

- बिग डेटा एनालिटिक्स संगठनों को अपनी जानकारी को बेहतर तरीके से संसाधित करने और उन क्षेत्रों की बेहतर समझ हासिल करने में सक्षम बनाता है जिनमें उन्हें सुधार करने की आवश्यकता है।



#### स्वचालन

- स्वचालन प्रक्रियाओं को गति देता है और व्यवसायों को अपने उद्देश्यों को अधिक कुशलता से प्राप्त करने की अनुमति देता है।



#### कृत्रिम होशियारी

- जबकि स्वचालन बढ़ रहा है, कृत्रिम बुद्धिमत्ता आखिरकार दिल के उजाले को देख रही है।
- हाल के वर्षों में कई नए AI माध्यम पेश किए गए हैं और उद्योग इन संसाधनों का विस्तार करने और अपने काम को अमल में लाने के तरीकों की तलाश कर रहे हैं।
- आर्टिफिशियल इंटेलिजेंस अब बड़े पैमाने पर लागू हो रहा है और आने वाले वर्षों में इसके जारी रहने की उम्मीद है।



#### स्मार्ट प्रौद्योगिकी

- हमारे जीवन को बेहतर बनाने के लिए काम करने वाले कई एकीकृत उपकरणों के परिणामस्वरूप घर अधिक स्मार्ट होते जा रहे हैं।
- एलेक्सा और अन्य पूरक संसाधन इस संबंध में बेहद उपयोगी साबित हुए हैं, जिससे घरों को प्रक्रिया में काफी अधिक कुशल और सुरक्षित बना दिया गया है।



#### अर्द्ध-ओटी नेटवर्क

- इंटरनेट ऑफ थिंग्स एक अवधारणा है जिसमें सभी डिजिटल उपकरण एक ही माध्यम से जुड़े हुए हैं, जिससे लोग अपने घरों में सब कुछ नियंत्रित कर सकते हैं।
- अधिक से अधिक ब्रांड यह महसूस कर रहे हैं कि यह भविष्य का तरीका है और यह तकनीकी रूप से संभव है।
- अधिक ब्रांड इस अवधारणा को शामिल करना शुरू कर रहे हैं, और इस विद्युत का समर्थन करने वाले आंकड़े इस क्षेत्र में सकारात्मक वृद्धि की गूँटि बताते हैं।



#### अप्रत्यासी वास्तविकता

- गेमिंग उद्योग हमेशा स्वयं प्रौद्योगिकी के क्षेत्र के साथ विगलित हुआ है, और आभासी वास्तविकता ने इसे एक बटन आगे बढ़ाया है, जिससे खेड़बने को डिजिटल अनुभव का लिखर मिलता है।
- गद्युअल रियलिटी गेमिंग ने नई तकनीक के परिणामस्वरूप लोकप्रियता दृशित कथना शुरू कर दिया है जो उद्योग की भारत में सुधार करता है।



#### सर्वापित वास्तविकता

- ओगमेंटेड रियलिटी कृत्रिम अनुभवों के लिए एक और दृष्टिकोण है, जिसके लिए अब लोगों की कटुप है।
- इसने जिस तरह से क्षेत्र को विकसित करने में सक्षम किया है, उसमें सुधार हुआ है।
- सर्वापित वास्तविकता गेमिंग उद्योग के बाहर बहुत अधिक पर्योचयता देख रही है और आभासी वास्तविकता की तुलना में अधिक कार्यप्रवणता देख रही है।



#### क्लाउड घेन डेटा

- पर्यायवाचीय का उपयोग लेन देन सक्ती डेटा को डिजिटल लेजर के रूप में सपहीत करने के लिए किया जाता है।
- हस्तासहित क्लाउड के रूप में सपहीत जानकारी अपरिवर्तनीय, परस्पर डेटा प्रविष्टिओं की एक सुबलत बनाने के लिए एक साथ जुड़ी हुई है।
- पर्यायवाचीय तकनीक अब केवल लोकप्रियता प्राप्त कर रही है और दुनिया भर के उद्योगों द्वारा इसे प्रदान किए जाने वाले तथ्यों के लिए अपनाया जा रहा है।



#### साइबर सुरक्षा

- लोगों को जिन समर्पित खतरों का सामना करना पड़ सकता है, वे केवल उन्नत डिजिटल शोधिया और प्रौद्योगिकी के विकास के साथ बढ़ रहे हैं।
- नतीजतन, तेजी से आर्थिक विस्तार को बनाए रखने के लिए हाल के वर्षों में साइबर सुरक्षा को काफी विस्तार करना पड़ा है।
- दुनिया भर के उद्योग साइबर सुरक्षा में निवेश के महत्व को पहचानते हैं। इतनी तेजी से क्षेत्र का विस्तार हो रहा है।



#### भविष्य कथनेवाला विश्लेषिकी कायांचयन

- पेंडिन्टिव एनालिटिक्स किसी दिशा के समर्पित परिणामों के बारे में अचिन्तयवाणी करने के लिए नई ज्ञान में डेटा का विश्लेषण करने की प्रक्रिया है।
- बाजार विश्लेषक इसे ब्रांडों के लिए यह निर्धारित करने के लिए एक अवसरदास्त मूल्यकाल उपकरण के रूप में देखते हैं कि उन्हें एक निश्चित दिशा में आगे बढ़ना चाहिए या नहीं।
- विश्लेषण की यह विधि अत्यंत कुशल साबित हुई है, जिससे उद्योगों को इस प्रक्रिया में महत्वपूर्ण ज्ञान में धन की कथत हुई है।



#### क्लाउड माइग्रेशन

- क्लाउड माइग्रेशन डेटा, एप्लिकेशन या अन्य व्यावसायिक तत्वों को क्लाउड कंप्यूटिंग वातावरण में स्थानांतरित करने की प्रक्रिया है।
- क्लाउड माइग्रेशन उन व्यवसायों के लिए अत्यधिक फायदेमंद साबित हुआ है जो डिजिटल होना चाहते हैं और अपने डेटा का बेहतर रिकॉर्ड रखना चाहते हैं।



#### क्वांटम कंप्यूटिंग अनुप्रयोग

- क्वांटम कंप्यूटिंग कई जटिल कार्यों को करने या समान दक्षता के साथ बड़ी मात्रा में डेटा को संसाधित करने के लिए जटिल समीकरणों और कार्यप्रणाली का उपयोग करने की प्रक्रिया है।
- यह उद्योगों की एक विस्तृत श्रृंखला के लिए बेहद फायदेमंद साबित हुआ है, यही वजह है कि यह तेजी से विकास का अनुभव कर रहा है।



#### ओपन सोर्स सॉल्यूशंस

- ओपन सोर्स प्रोग्राम उपयोगकर्ताओं को प्रोग्राम में कुछ सबसे महत्वपूर्ण फाइलों और ढांचे तक पहुंच प्रदान करते हैं, जिससे वे इसे आसानी से संशोधित कर सकते हैं।
- जैसे-जैसे अधिक उपयोगकर्ता तकनीकी दक्षता हासिल करते हैं, उन्हें सीधे अनुप्रयोगों के साथ काम करने का विकल्प देना बेहद फायदेमंद साबित हो रहा है।



#### एज कंप्यूटिंग

- एज कंप्यूटिंग तब होती है जब बड़ी मात्रा में डेटा को नेटवर्क के किनारे के पास संसाधित किया जाता है, न कि जहां डेटा उत्पन्न होता है।
- यह प्रसंस्करण की दक्षता और अनुकूलन को बढ़ाने में मदद करता है।
- अधिक विकास माध्यम इसकी दक्षता के कारण पारंपरिक पर इस प्रकार की कंप्यूटिंग के लिए चयन कर रहे हैं।
- यह यह भी दर्शाता है कि अधिक कंप्यूटिंग विधियों और कार्यक्रम निर्माण को विकसित किया जा रहा है।



#### चैटबॉट्स

- चैटबॉट ऐसे प्रोग्राम हैं जो विशिष्ट प्रश्नों का विशिष्ट तरीके से जवाब देते हैं और ग्राहकों को कुछ अधिक बुनियादी कार्यों में सहायता करते हैं जिनकी उन्हें आवश्यकता हो सकती है।
- बेशक, वे अभी भी रीयल-टाइम ग्राहक सेवा के लिए पूर्ण विकल्प नहीं हैं, जिसने उन्हें विकास में बने रहने की अनुमति दी है।

चित्र 1.10 आईटी-आईटीईएस उद्योग में प्रमुख उभरते रुझान

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दें-

आईटी उद्योग की विभिन्न विशेषताओं की सूची बनाएं।

आईटी उद्योग में उपयोग किए जाने वाले विभिन्न वितरण मॉडलों की व्याख्या करें।

आईटीईएस उद्योग के विभिन्न उप-क्षेत्रों की व्याख्या कीजिए।

आईटी सेवा क्षेत्र द्वारा प्रदान की जाने वाली सेवाओं की व्याख्या करें।

व्यवसाय प्रक्रिया प्रबंधन से आप क्या समझते हैं?

आईटी उद्योग में केपीओ और एलपीओ के बीच अंतर करें।

आईटी उद्योग में प्रमुख उभरती प्रवृत्तियों पर चर्चा करें।

एमसीक्यू/निम्नलिखित का मिलान करें-

बीपीओ डेस्टिनेशन तय करते समय कौन से कारक प्रभावित होते हैं?

विश्वसनीय दूरसंचार अवसंरचना

श्रम की लागत

मजबूत नियामक ढांचा

उपरोक्त सभी

रिक्त स्थान भरें

\_\_\_\_\_ औद्योगिक दुनिया के सबसे बड़े क्षेत्रों में से एक है।

ज्ञान प्रक्रिया आउटसोर्सिंग (केपीओ) में \_\_\_\_\_ का एक घटक शामिल है।

\_\_\_\_\_ जो सेवा की गुणवत्ता में सुधार करके व्यवसाय को सक्षम बनाता है वह है आईटी सक्षम सेवाएं।

सेवा क्षेत्र में सूचना प्रौद्योगिकी के उपयोग से \_\_\_\_\_ में सुधार होता है और पारदर्शिता बढ़ती है।

## यूनिट 1.2: एक सॉफ्टवेयर डेवलपर के लिए करियर पथ

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. एक सॉफ्टवेयर डेवलपर के लिए करियर पथ की पहचान करने में

### 1.2.1 सॉफ्टवेयर विकास

"सॉफ्टवेयर डेवलपमेंट " सॉफ्टवेयर बनाने, डिजाइन करने और समर्थन करने की प्रक्रिया से संबंधित कंप्यूटर विज्ञान गतिविधियों के एक समूह को संदर्भित करता है। मुख्य बात यह है कि काम केवल कोडर्स या विकास दल तक ही सीमित नहीं है। कार्यरत व्यक्ति कार्य के हिस्से के रूप में, वैज्ञानिक, डिवाइस फैब्रिकेटर और हार्डवेयर डिजाइनर कोड और एल्गोरिदम बनाते हैं। सॉफ्टवेयर डेवलपर्स द्वारा किया गया कार्य संबंधित है लेकिन सॉफ्टवेयर इंजीनियरों से अलग है। सॉफ्टवेयर इंजीनियर आमतौर पर विशिष्ट कार्यों के लिए सॉफ्टवेयर बनाने के लिए इंजीनियरिंग सिद्धांतों का उपयोग करते हैं। सॉफ्टवेयर डेवलपर संपूर्ण सॉफ्टवेयर डेवलपर जीवनचक्र में अधिक शामिल होते जा रहे हैं। एक अन्य अंतर कस्टम सॉफ्टवेयर डेवलपर और वाणिज्यिक सॉफ्टवेयर डेवलपर के बीच का अंतर है। कस्टम सॉफ्टवेयर विकास विशिष्ट उद्देश्यों के लिए अनुप्रयोगों के निर्माण को संदर्भित करता है, आमतौर पर संगठनों के भीतर।

सॉफ्टवेयर उत्पादों के रूप में, वाणिज्यिक सॉफ्टवेयर डेवलपर उपभोक्ताओं के एक बड़े समूह को सॉफ्टवेयर वितरित करता है। कंप्यूटर प्रोग्राम के पीछे रचनात्मक दिमाग को सॉफ्टवेयर डेवलपर्स के रूप में जाना जाता है। कुछ प्रोग्रामर ऐसे एप्लिकेशन बनाते हैं जो लोगों को कंप्यूटर या अन्य डिवाइस पर विशिष्ट कार्य करने की अनुमति देते हैं। अन्य अंतर्निहित सिस्टम बनाते हैं जो पावर डिवाइस या नेटवर्क को नियंत्रित करते हैं।

### 1.2.2 सॉफ्टवेयर डेवलपर्स की भूमिकाएं और जिम्मेदारियां

सॉफ्टवेयर डेवलपर्स की निम्नलिखित भूमिकाएँ और जिम्मेदारियाँ निम्न हैं:



चित्र 1.11 सॉफ्टवेयर डेवलपर की भूमिकाएं और जिम्मेदारियां

### 1.2.3 सॉफ्टवेयर डेवलपर्स की प्रमुख दक्षताएं

सॉफ्टवेयर डेवलपर्स की प्रमुख दक्षताएं निम्नलिखित हैं।

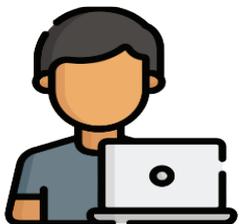
<p><b>विवेचनात्मक कौशल</b></p> <ul style="list-style-type: none"> <li>उपयोगकर्ताओं की जरूरतों का विवेक्षण करें और फिर उन जरूरतों को पूरा करने के लिए सॉफ्टवेयर डिजाइन करें</li> </ul>	<p><b>संचार कौशल</b></p> <ul style="list-style-type: none"> <li>किसी प्रोजेक्ट पर काम करने वाले अन्य लोगों को स्पष्ट निर्देश देने में सक्षम</li> <li>अपने ग्राहकों को समझाएं कि सॉफ्टवेयर कैसे काम करता है और किसी भी प्रश्न का उत्तर देता है</li> </ul>	<p><b>रचनात्मकता</b></p> <ul style="list-style-type: none"> <li>अपने रचनात्मक विचारों के आधार पर नए कंप्यूटर सॉफ्टवेयर का निर्माण करना</li> </ul>
<p><b>विस्तार उन्मुख</b></p> <ul style="list-style-type: none"> <li>ध्यान केंद्रित करने और विस्तार पर ध्यान देने में सक्षम</li> </ul>	<p><b>पारस्परिक कौशल</b></p> <ul style="list-style-type: none"> <li>सफल सॉफ्टवेयर की डिजाइनिंग, विकास और प्रोयामिंग में योगदान करने वाले अन्य लोगों के साथ अच्छी तरह से काम करने में सक्षम</li> </ul>	<p><b>समस्या समाधान करने की कुशलताएं</b></p> <ul style="list-style-type: none"> <li>डिजाइन प्रक्रिया के दौरान उत्पन्न होने वाली समस्याओं को हल करने में सक्षम।</li> </ul>

चित्र 1.12 सॉफ्टवेयर डेवलपर की प्रमुख योग्यताएं

### 1.2.4 सॉफ्टवेयर डेवलपमेंट में कैरियर के अवसर

सॉफ्टवेयर डेवलपमेंट के क्षेत्र में अवसर प्रौद्योगिकी की प्रगति के साथ लॉकस्टेप में विस्तार कर रहे हैं। कुशल प्रोग्रामर के लिए प्रबंधन पद उपलब्ध हैं जो पारस्परिक संबंध और सॉफ्ट कौशल विकसित करना चाहते हैं। एक तकनीकी रूप से कुशल कम्युनिकेटर के पास कई विकल्प होते हैं और अंततः वरिष्ठ डेवलपर, टीम लीड, आर्किटेक्ट और मैनेजर जैसी कई भूमिकाओं के लिए अर्हता प्राप्त कर सकते हैं, साथ ही वरिष्ठ प्रबंधन में कदम रख सकते हैं। विभिन्न कारकों के आधार पर उच्चतम-भुगतान वाले वरिष्ठ पदों का मार्ग कई महीनों से लेकर कई वर्षों तक का समय ले सकता है। कैरियर पथ की प्रगति की दर मुख्य रूप से प्रदर्शित तकनीकी क्षमता, आंतरिक ड्राइव, महत्वपूर्ण सोच और संचार कौशल द्वारा निर्धारित की जाती है।

एक विशिष्ट सॉफ्टवेयर डेवलपर कैरियर पथ:

<p>जूनियर डेवलपर</p> 	<p>यह प्रवेश-स्तर या शुरुआती स्थिति है कि एक डेवलपर के पास वास्तविक दुनिया के कंप्यूटर प्रोग्रामिंग अनुभव के साथ कॉलेज से बाहर या वोज यू (Woz U) सॉफ्टवेयर डेवलपर जैसे त्वरित पाठ्यक्रम होगा।</p> <p>एक जूनियर डेवलपर के लिए विशिष्ट आवश्यकताओं में शामिल हैं:</p> <ul style="list-style-type: none"> <li>सरल स्क्रिप्ट लिखने की क्षमता</li> <li>अपेक्षित अनुप्रयोग जीवनचक्र की समझ</li> <li>डेटाबेस और एप्लिकेशन सेवाओं की बुनियादी समझ, जैसे कि क्यू और कैशिंग</li> </ul> <p>हालांकि, जूनियर और सीनियर डेवलपर के बीच मुख्य अंतर अनुभव है।</p>
--	---

सॉफ्टवेयर डेवलपर या वरिष्ठ सॉफ्टवेयर डेवलपर



एक सॉफ्टवेयर डेवलपर, जिसे एक वरिष्ठ सॉफ्टवेयर डेवलपर के रूप में भी जाना जाता है, वह है जिसने संपूर्ण अनुप्रयोगों के निर्माण में महारत हासिल की है। एक सॉफ्टवेयर डेवलपर के करियर का अधिकांश हिस्सा एक वरिष्ठ डेवलपर के रूप में व्यतीत किया जाएगा।

एक सॉफ्टवेयर डेवलपर के लिए विशिष्ट आवश्यकताओं में शामिल हैं:

- कई वर्षों के पेशेवर प्रोग्रामिंग अनुभव के साथ
- जटिल कोड लिखने की क्षमता
- डेटाबेस, एप्लिकेशन सेवाओं और एप्लिकेशन जीवनचक्र विकास की गहन समझ रखने वाले

लीड डेवलपर



लीड डेवलपर या आर्किटेक्ट वह है जो अधिक जिम्मेदारी और चुनौती चाहता है लेकिन एक टीम का प्रबंधन नहीं करना चाहता। वे विशिष्ट वरिष्ठ डेवलपर्स भी हैं जिनके लिए अन्य प्रोग्रामर मार्गदर्शन की तलाश करते हैं। कोड लिखते समय, लीड डेवलपर्स आमतौर पर काम का समन्वय करते हैं और निर्णयों को लागू करते हैं। उनकी प्राथमिक जिम्मेदारी अन्य डेवलपर्स द्वारा कार्यान्वित जटिल प्रणालियों को डिजाइन करना है।

लीड डेवलपर पदों के लिए विशिष्ट आवश्यकताओं में शामिल हैं:

- कई वर्षों का पेशेवर प्रोग्रामिंग अनुभव
- विषय वस्तु विशेषज्ञता प्रदर्शित करता है
- वही तकनीकी कौशल रखना जो एक वरिष्ठ डेवलपर के पास हो सकता है
- जटिल समस्याओं को हल करने के लिए सॉफ्टवेयर की अवधारणा, योजना और निर्माण करने की क्षमता

लीड डेवलपर पदों को अक्सर मध्य-स्तरीय प्रबंधन पदों पर कदम रखने वाले पत्थरों के रूप में देखा जाता है, जबकि आर्किटेक्ट एक तकनीकी स्थिति है जो शायद ही कभी एक कदम है। तकनीकी वास्तुकारों को अक्सर तकनीकी कैरियर की सीढ़ी के शिखर के रूप में माना जाता है।

डेवलपमेंट टीम लीड या सॉफ्टवेयर डेवलपमेंट मैनेजर



एक सॉफ्टवेयर डेवलपर के करियर में प्रबंधन अगला तार्किक और पारंपरिक कदम है। प्रबंधकों के पास विभिन्न फोकस और जिम्मेदारियां हैं, जिनमें शामिल हैं:

- बड़े पैमाने पर परियोजनाओं का प्रबंधन
- डेवलपर्स की टीमों का प्रबंधन
- डेवलपर्स को काम पर रखना और फायरिंग करना

मध्य-स्तरीय प्रबंधक कार्यप्रवाह और उत्पादकता की निगरानी करता है और उच्च-स्तरीय नेता या प्रबंधक को रिपोर्ट करता है। उनका विशिष्ट कार्य विकास दल के साथ एप्लिकेशन और प्रोजेक्ट लीडर की जरूरतों का समन्वय करना है। इस स्थिति के लिए उत्कृष्ट पारस्परिक कौशल और संघर्षों को प्रभावी ढंग से हल करने की क्षमता की आवश्यकता होती है। इसमें आवश्यकतानुसार टीम के सदस्यों को हटाना और जोड़ना भी शामिल है। मिड-लेवल मैनेजर बनने का एक बड़ा नुकसान यह है कि उन्हें लोगों को नौकरी से निकालना पड़ सकता है। बहुत से लोग इसे एक मनोवैज्ञानिक और भावनात्मक चुनौती मानते हैं, लेकिन जब इसमें महारत हासिल हो जाती है, तो यह एक शक्तिशाली नेतृत्व कौशल हो सकता है।

निदेशक या उपाध्यक्ष या मुख्य प्रौद्योगिकी अधिकारी



एक वरिष्ठ और मध्य स्तर के प्रबंधक के बीच प्राथमिक अंतर यह है कि वरिष्ठ प्रबंधन आम तौर पर अन्य प्रबंधकों के काम की निगरानी करता है और रणनीति और कॉर्पोरेट दिशा निर्धारित करने के लिए जिम्मेदार हो सकता है। वरिष्ठ प्रबंधन पदों में निम्नलिखित शामिल हैं:

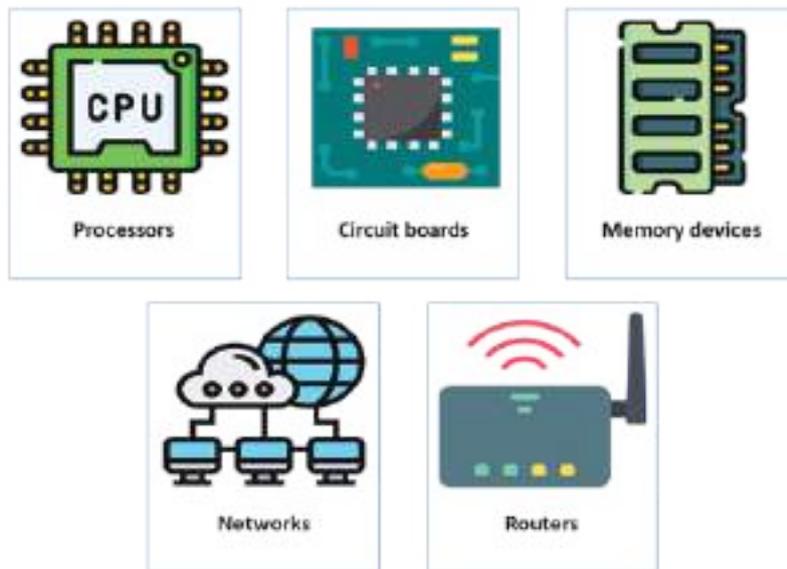
- निर्देशक
- उपाध्यक्ष
- मुख्य तकनीकी अधिकारी

एक वरिष्ठ प्रबंधक का काम कार्यकारी स्तर के निर्णय लेना है जैसे कि दीर्घकालिक रणनीति, लक्ष्य निर्धारण और संपूर्ण विभागों के लिए जवाबदेही। उन्हें अपने प्रबंधकों को विशेष रूप से मापने योग्य उद्देश्यों के लिए भी प्रेरित करना चाहिए और उन्हें जवाबदेह ठहराना चाहिए। वरिष्ठ स्तर के अधिकारी मध्य स्तर के प्रबंधकों को काम पर रखते हैं, पर्यवेक्षण करते हैं और आग लगाते हैं और केवल अन्य वरिष्ठ स्तर के अधिकारियों, जैसे सीईओ या निदेशक मंडल को रिपोर्ट करते हैं।

चित्र 1.1 सॉफ्टवेयर डेवलपर का करियर पथ

### 1.2.5 सॉफ्टवेयर डेवलपर से संबंधित अन्य व्यावसायिक करियर

1. कंप्यूटर और सूचना अनुसंधान वैज्ञानिक - कंप्यूटर और सूचना शोधकर्ता कंप्यूटिंग प्रौद्योगिकी के लिए नए दृष्टिकोणों का आविष्कार और डिजाइन करते हैं, साथ ही मौजूदा प्रौद्योगिकी के लिए नए अनुप्रयोगों की खोज करते हैं। वे व्यवसाय, चिकित्सा, विज्ञान और अन्य क्षेत्रों में जटिल कंप्यूटिंग समस्याओं का शोध और समाधान करते हैं।
2. कंप्यूटर और सूचना प्रणाली प्रबंधक - कंप्यूटर और सूचना प्रणाली प्रबंधक, जिन्हें सूचना प्रौद्योगिकी (आईटी) प्रबंधकों या आईटी परियोजना प्रबंधकों के रूप में भी जाना जाता है, एक संगठन की कंप्यूटर से संबंधित गतिविधियों की योजना, समन्वय और निर्देशन करते हैं। वे एक संगठन के सूचना प्रौद्योगिकी लक्ष्यों को निर्धारित करने में सहायता करते हैं और उन लक्ष्यों को प्राप्त करने के लिए कंप्यूटर सिस्टम को लागू करने के प्रभारी हैं।
3. कंप्यूटर हार्डवेयर इंजीनियर - कंप्यूटर हार्डवेयर इंजीनियर कंप्यूटर सिस्टम और घटकों के अनुसंधान, डिजाइन, डेवलपर और परीक्षण के लिए जिम्मेदार होते हैं जैसे:



चित्र 1.13 कंप्यूटर हार्डवेयर घटक

4. कंप्यूटर प्रोग्रामर: कंप्यूटर प्रोग्रामर कोड बनाते हैं और परीक्षण करते हैं जो कंप्यूटर एप्लिकेशन और सॉफ्टवेयर प्रोग्राम को ठीक से काम करने में सक्षम बनाता है। वे सॉफ्टवेयर डेवलपर्स और इंजीनियरों द्वारा बनाए गए प्रोग्राम डिजाइन को कंप्यूटर-पठनीय निर्देशों में परिवर्तित करते हैं।



Fig 1.14 कंप्यूटर प्रोग्रामर

5. कंप्यूटर नेटवर्क आर्किटेक्ट्स: कंप्यूटर नेटवर्क आर्किटेक्ट लोकल एरिया नेटवर्क (LAN), वाइड एरिया नेटवर्क (WAN), और इंटरनेट जैसे डेटा संचार नेटवर्क की योजना बनाते हैं और उनका निर्माण करते हैं। ये नेटवर्क दो-कार्यालय कनेक्शन से लेकर अगली पीढ़ी की नेटवर्किंग क्षमताओं जैसे क्लाउड इन्फ्रास्ट्रक्चर तक होते हैं जो कई ग्राहकों को सेवा प्रदान करते हैं।



Fig 1.15 कंप्यूटर नेटवर्क आर्किटेक्ट्स:

6. कंप्यूटर सहायता विशेषज्ञ: कंप्यूटर सहायता विशेषज्ञ कंप्यूटर उपयोगकर्ताओं और व्यवसायों की सहायता और सलाह देते हैं। ये पेशेवर या तो कंप्यूटर नेटवर्क का समर्थन करते हैं या कंप्यूटर उपयोगकर्ताओं को सीधे तकनीकी सहायता प्रदान करते हैं।



Fig 1.16 कंप्यूटर सहायता विशेषज्ञ

7. कंप्यूटर सिस्टम विश्लेषक: कंप्यूटर सिस्टम विश्लेषक, जिन्हें सिस्टम आर्किटेक्ट के रूप में भी जाना जाता है, संगठन के वर्तमान कंप्यूटर सिस्टम और प्रक्रियाओं का मूल्यांकन और अध्ययन करते हैं और संगठन को अधिक कुशलता और प्रभावी ढंग से संचालित करने में मदद करने के लिए समाधान तैयार करते हैं। वे अपनी संबंधित जरूरतों और सीमाओं को समझकर व्यापार और सूचना प्रौद्योगिकी (आईटी) को जोड़ते हैं।



चित्र 1.17 कंप्यूटर सिस्टम विश्लेषक

8. डेटाबेस एडमिनिस्ट्रेटर: डेटाबेस एडमिनिस्ट्रेटर (डीबीए) विशेष सॉफ्टवेयर का उपयोग करके वित्तीय जानकारी और ग्राहक शिपिंग रिकॉर्ड जैसे डेटा को स्टोर और व्यवस्थित करते हैं। वे सुनिश्चित करते हैं कि डेटा उपयोगकर्ताओं के लिए सुलभ है और अनधिकृत पहुंच से सुरक्षित है।



Fig 1.18 डेटाबेस एडमिनिस्ट्रेटर:

9. सूचना सुरक्षा विश्लेषक: सूचना सुरक्षा विश्लेषक कंपनी के कंप्यूटर नेटवर्क और सिस्टम की सुरक्षा के लिए सुरक्षा उपाय तैयार करते हैं और लागू करते हैं। जैसे-जैसे साइबर हमलों की संख्या बढ़ेगी, वैसे-वैसे उनकी जिम्मेदारियां भी बढ़ेंगी।



चित्र 1.19 सूचना सुरक्षा विश्लेषक

10. गणितज्ञ और सांख्यिकीविद: गणितज्ञ और सांख्यिकीविद डेटा का विश्लेषण करते हैं और व्यवसाय, इंजीनियरिंग, स्वास्थ्य देखभाल और अन्य क्षेत्रों में वास्तविक दुनिया की समस्याओं को हल करने के लिए गणितीय और सांख्यिकीय तकनीकों का उपयोग करते हैं।



चित्र 1.20 गणितज्ञ और सांख्यिकीविद

11. उत्तर माध्यमिक शिक्षक: वे हाई स्कूल स्तर से परे विभिन्न प्रकार के शैक्षणिक और तकनीकी विषयों में छात्रों को निर्देश देते हैं। वे शोध भी कर सकते हैं और विद्वानों के कागजात और किताबें प्रकाशित कर सकते हैं।



चित्र 1.21 उत्तर माध्यमिक शिक्षक:

12. वेब डेवलपर: वेब डेवलपर वेबसाइट बनाते और डिजाइन करते हैं। वे साइट की उपस्थिति के प्रभारी हैं। वे साइट के तकनीकी पहलुओं के प्रभारी भी हैं, जैसे प्रदर्शन और क्षमता, जो वेबसाइट की गति और यातायात को संभालने की क्षमता के उपाय हैं। वेब डेवलपर साइट के लिए सामग्री भी बना सकते हैं।



चित्र 1.22 . वेब डेवलपर

## सारांश



- सूचना प्रौद्योगिकी एसोसिएशन ऑफ अमेरिका (आईटीए) "सूचना प्रौद्योगिकी" को "कंप्यूटर आधारित सूचना प्रणाली के सभी संभावित पहलुओं" के रूप में परिभाषित करता है। कंप्यूटर सिस्टम से लेकर सॉफ्टवेयर और हार्डवेयर के डिजाइन, कार्यान्वयन, अध्ययन और विकास तक सब कुछ आईटी उद्योग में शामिल है।
- सूचना प्रौद्योगिकी उद्योग के लिए बड़े पैमाने की अर्थव्यवस्थाएं उच्च हैं। अतिरिक्त सॉफ्टवेयर या हार्डवेयर की प्रत्येक इकाई की सीमांत लागत इसके परिणामस्वरूप होने वाले मूल्यवर्धन की तुलना में नगण्य है।
- आईटी उद्योग ई-गवर्नेंस के माध्यम के रूप में काम कर सकता है, क्योंकि यह सूचना तक आसान पहुंच का आश्वासन देता है। सेवा क्षेत्र में सूचना प्रौद्योगिकी के उपयोग से परिचालन दक्षता में सुधार होता है और पारदर्शिता बढ़ती है। यह कौशल निर्माण के माध्यम के रूप में भी कार्य करता है।
- सूचना प्रौद्योगिकी सेवाओं जैसे परीक्षण या गुणवत्ता आश्वासन, अनुप्रयोग विकास, और रखरखाव, और डेटा केंद्र संचालन को कम परिचालन लागत और उच्च दक्षता के उद्देश्य से तीसरे पक्ष के विक्रेता/कंपनी को आउटसोर्स किया जाता है। जेनपैकट, विप्रो, टीसीएस और इंफोसिस बीपीएम क्षेत्र में कुछ प्रमुख आउटसोर्सिंग सेवा प्रदाता हैं।
- उपभोक्ता व्यवहार का ज्ञान समान रूप से वांछित कौशल है। बीपीएम क्षेत्र में काम करने वाले लोगों को ग्राहकों की आवश्यकताओं को सुनने और उसके अनुसार कार्य करने का धैर्य रखना चाहिए।
- इंजीनियरिंग सर्विसेज आउटसोर्सिंग (ईएसओ) औद्योगिक दुनिया के सबसे बड़े क्षेत्रों में से एक है और इसके 2020 तक \$40 बिलियन तक बढ़ने का अनुमान है। लागत और गुणवत्ता इंजीनियरिंग सेवाओं को आउटसोर्स करने के दो मुख्य कारण हैं। बीपीएम क्षेत्र के विपरीत, जहां सेवाओं को आउटसोर्स करने के लिए कम परिचालन लागत एक प्रमुख कारक है, ईएसओ एक डोमेन-केंद्रित कंपनी / तीसरे पक्ष से बेहतर गुणवत्ता वाले उत्पाद प्राप्त करने की क्षमता पर केंद्रित है।
- इंजीनियरिंग सेवाओं के लिए तकनीकी कौशल और अच्छे इंजीनियरिंग निर्णय की आवश्यकता होती है। इसलिए डोमेन विशेषज्ञता एक जरूरी है।
- ज्ञान प्रक्रिया आउटसोर्सिंग (केपीओ) में बीपीएम, अनुसंधान और विश्लेषण का एक घटक शामिल है। ज्ञान प्रसंस्करण में शामिल एक कंपनी इस प्रकार एक ऐसा संगठन होगा जहां उद्देश्य प्रक्रियाओं को पूरा करने वाले लोगों के डोमेन ज्ञान, कौशल और अनुभव पर निर्भर होते हैं।
- "सॉफ्टवेयर डेवलपर कंप्यूटर विज्ञान गतिविधियों के एक सेट को संदर्भित करता है जो सॉफ्टवेयर बनाने, डिजाइन करने, तैनात करने और समर्थन करने की प्रक्रिया के लिए समर्पित है।"
- सॉफ्टवेयर डेवलपर्स कंप्यूटर प्रोग्राम के पीछे रचनात्मक दिमाग हैं।
- विकासकर्ता जो किसी सॉफ्टवेयर परियोजना का नियोजन चरणों से कार्यान्वयन के माध्यम से पर्यवेक्षण करते हैं, उन्हें कभी-कभी सूचना प्रौद्योगिकी (आईटी) परियोजना प्रबंधक कहा जाता है। ये कार्यकर्ता यह सुनिश्चित करने के लिए परियोजना की प्रगति की निगरानी करते हैं कि यह समय सीमा, मानकों और लागत लक्ष्यों को पूरा करता है। आईटी परियोजना प्रबंधक जो किसी संगठन के आईटी विभाग या आईटी नीतियों की योजना और निर्देशन करते हैं, उन्हें कंप्यूटर और सूचना प्रणाली प्रबंधकों के प्रोफाइल में शामिल किया जाता है।
- एप्लिकेशन सॉफ्टवेयर डेवलपर उपभोक्ताओं के लिए कंप्यूटर एप्लिकेशन, जैसे वर्ड प्रोसेसर और गेम डिजाइन करते हैं। वे आम जनता को बेचे जाने वाले विशिष्ट ग्राहक या वाणिज्यिक सॉफ्टवेयर के लिए कस्टम सॉफ्टवेयर बना सकते हैं।
- डेवलपर्स को किसी प्रोजेक्ट पर काम करने वाले अन्य लोगों को स्पष्ट निर्देश देने में सक्षम होना चाहिए। उन्हें अपने ग्राहकों को यह भी बताना चाहिए कि सॉफ्टवेयर कैसे काम करता है और किसी भी प्रश्न का उत्तर देता है।
- समस्या समाधान करने की कुशलताएं। चूंकि डेवलपर्स शुरू से अंत तक सॉफ्टवेयर के प्रभारी होते हैं, इसलिए उन्हें पूरी डिजाइन प्रक्रिया में आने वाली समस्याओं को हल करने में सक्षम होना चाहिए।
- एक सॉफ्टवेयर डेवलपर, या वरिष्ठ सॉफ्टवेयर डेवलपर वह है जो संपूर्ण एप्लिकेशन बनाने में कुशल हो गया है। एक सॉफ्टवेयर डेवलपर के करियर पथ में एक वरिष्ठ-डेवलपर के रूप में कई साल खर्च करना शामिल होगा।
- लीड डेवलपर या आर्किटेक्ट की भूमिका किसी ऐसे व्यक्ति के लिए होती है जो अधिक जिम्मेदारी और चुनौती चाहता है लेकिन टीम के प्रबंधन में दिलचस्पी नहीं रखता है।
- प्रबंधन सॉफ्टवेयर डेवलपर कैरियर पथ के साथ अगला तार्किक और पारंपरिक कदम है।
- एक वरिष्ठ और मध्य स्तर के प्रबंधक के बीच मुख्य अंतर यह है कि वरिष्ठ प्रबंधन अन्य प्रबंधकों के काम की देखरेख करता है और रणनीति और कॉर्पोरेट दिशा निर्धारित करने के लिए कुछ जिम्मेदारी हो सकती है।

## अभ्यास



1. रिक्त स्थान भरें:

1. \_\_\_\_\_ वह व्यक्ति है जो संपूर्ण एप्लिकेशन बनाने में कुशल हो गया है।
2. \_\_\_\_\_ सॉफ्टवेयर डेवलपर करियर पथ के साथ अगला तार्किक और पारंपरिक कदम है।
3. नॉलेज प्रोसेस आउटसोर्सिंग (केपीओ) में \_\_\_\_\_ का एक घटक शामिल है।
4. \_\_\_\_\_ जो किसी संगठन के आईटी विभाग या आईटी नीतियों की योजना और निर्देशन करते हैं, उन्हें कंप्यूटर और सूचना प्रणाली प्रबंधकों के प्रोफाइल में शामिल किया जाता है।

निम्नलिखित प्रश्नों के उत्तर दो-

1. सिस्टम सॉफ्टवेयर डेवलपर्स से आप क्या समझते हैं?
2. सॉफ्टवेयर डेवलपर के विभिन्न चरणों की व्याख्या कीजिए।
3. सॉफ्टवेयर डेवलपर्स के लिए महत्वपूर्ण गुणों की सूची बनाएं।
4. सॉफ्टवेयर डेवलपर्स से संबंधित करियर की सूची बनाएं।



## 2. सॉफ्टवेयर डेवलपमेंट का परिचय



IT - ITeS SSC  
NASSCOM

यूनिट 2.1 सॉफ्टवेयर डेवलपमेंट और आईटी सर्विस



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. व्यवसाय में आईटी सॉफ्टवेयर सेवा क्षेत्र के महत्व पर चर्चा करने में

## यूनिट 2.1: सॉफ्टवेयर विकास और आईटी सेवाएं

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. आईटी-सक्षम सेवाओं की स्थापना में उच्च पूंजी निवेश, निरंतर व्यावसायिक उपलब्धता, टर्नअराउंड समय, संचार लागत आदि के परिनियोजन मुद्दों की सूची बनाने में

### 2.1.1 आईटी सॉफ्टवेयर सेवा क्षेत्र का महत्व

हर व्यवसाय, चाहे वह बड़ा हो या छोटा, को उत्पादकता बढ़ाने के लिए प्रभावी आईटी सेवाओं और प्रणालियों की आवश्यकता होती है। मजबूत और भरोसेमंद आईटी सेवाएं संगठनों को प्रतिस्पर्धी बने रहने में सक्षम बनाती हैं, चाहे वह इन-हाउस सूचना प्रौद्योगिकी विशेषज्ञों को नियुक्त करके या आउटसोर्सिंग तकनीकी सहायता के लाभों को अधिकतम करके। नतीजतन, संगठनों के लिए नवीनतम तकनीकी रुझानों पर अद्यतित रहना और अपने मौजूदा कंप्यूटर सिस्टम को लगातार अपडेट करने पर ध्यान केंद्रित करना महत्वपूर्ण है।

सूचना प्रौद्योगिकी वाणिज्य और व्यावसायिक क्षेत्रों के विकास और विस्तार में सहायता करती है, जिसके परिणामस्वरूप अधिकतम उत्पादन होता है। सूचना प्रौद्योगिकी में प्रगति के कारण विभिन्न क्षेत्रों को व्यवसाय उत्पन्न करने में लगने वाला समय कम हो गया है। यह इलेक्ट्रॉनिक सुरक्षा, डेटा का भंडारण और प्रभावी संचार प्रदान करता है।

ग्राहक उन कंपनियों के साथ काम करना पसंद करते हैं जो तकनीकी रुझानों के साथ बनी रहती हैं। आईटी समर्थन सेवाओं के महत्व के बारे में पूरी तरह से जागरूक होना, विशेष रूप से इसकी संरचना में बड़े और छोटे परिवर्तन, साथ ही इसकी लागत और मांग, अधिक सुव्यवस्थित संचालन सुनिश्चित करते हुए समर्पण और निर्भरता को प्रदर्शित करता है। आईटी सेवाओं और प्रणालियों में समस्या-समाधान और रचनात्मक संसाधनों की एक विविध श्रेणी शामिल है। प्रत्येक सावधानीपूर्वक डिज़ाइन किया गया और उच्च तकनीक वाला उपकरण इस बात पर महत्वपूर्ण प्रभाव डालता है कि वे कंपनी की सेवाओं और आंतरिक कार्यों को कैसे करते हैं। निम्नलिखित सूची आईटी समर्थन सेवाओं और परिदृश्यों के महत्व और लाभों की गणना करती है जिसमें सूचना प्रौद्योगिकी कंपनी के सभी बड़े और छोटे निर्णयों से जुड़ी होती है:



चित्र 2.1 आईटी सॉफ्टवेयर सेवा क्षेत्र का महत्व

1. प्रभावी डेटा प्रबंधन- व्यवसाय कर्मचारी के वेतन, कमाई और एचआर रिकॉर्ड जैसी संवेदनशील जानकारी रखते हैं। नतीजतन, डेटा भंडारण और प्रबंधन किसी भी प्रकार के व्यवसाय के लिए महत्वपूर्ण हैं, और वे एक प्रमुख उदाहरण हैं कि आईटी सहायता की आवश्यकता क्यों है। डेटा प्रबंधन में सक्षम आईटी सेवाओं को शामिल करने के लिए व्यावसायिक आवश्यकताओं की अधिक गहन परीक्षा के साथ-साथ कंपनी के डेटा परिदृश्य की अधिक गहन परीक्षा की आवश्यकता होती है। सभी महत्वपूर्ण सूचनाओं और सॉफ्टवेयर के लिए एक अच्छी बैकअप रणनीति कंपनी को डेटा उल्लंघनों से बचने में मदद कर सकती है। एक प्रभावी डेटा प्रबंधन रणनीति तैयार करने के लिए कंपनी के मूल्यवान डेटा को संभालने और सुरक्षित करने के लिए जानकार आईटी कर्मचारियों के साथ एक उच्च कुशल टीम की आवश्यकता होती है। जब ऐसा होता है, तो संवेदनशील कंपनी और कर्मचारी डेटा को हैकर्स और इसे उजागर करने के अन्य प्रयासों से प्रभावी ढंग से सुरक्षित किया जाता है।
2. निर्णय लेने में सुधार - ठोस बाजार अनुसंधान ध्वनि व्यावसायिक निर्णयों की नींव है। इस प्रक्रिया में वीडियो कॉन्फ्रेंस, सोशल मीडिया पर सार्वजनिक टिप्पणियां, उद्योग मंच और ऑनलाइन सर्वेक्षण प्रतिक्रिया शामिल हैं। ये प्रक्रियाएं बेहतर व्यावसायिक निर्णय लेने और उद्देश्यों के निर्माण में सहायता करती हैं। कंपनियां Microsoft CRM Dynamics और Google Analytics जैसी डिजिटल मार्केटिंग तकनीकों का उपयोग करके विकास और विकास को ट्रैक कर सकती हैं। बड़े पैमाने पर, आईटी सॉफ्टवेयर मौलिक उद्देश्यों को प्राप्त करने के लिए अधिक सटीक और उन्नत विकल्प प्रदान करके मौजूदा रणनीतियों में सुधार करता है।
3. जटिल समस्याओं का समाधान - आईटी समर्थन के महत्व का एक अन्य ठोस उदाहरण कंपनी को चालू रखने वाली आंतरिक प्रणालियों को प्रभावित करने वाली जटिल चुनौतियों के लिए उन्नत और सटीक समाधानों का कार्यान्वयन है। आईटी सेवाएं और सिस्टम संगठनों को वे संसाधन प्रदान करते हैं जिनकी उन्हें हार्डवेयर को अपग्रेड करने की आवश्यकता होती है जैसे मेमोरी स्टोरेज, प्रोसेसर गति और प्रदर्शन गुणवत्ता। माइंड-मैपिंग सॉफ्टवेयर, सहयोगी सिस्टम, और काम को अधिक सुव्यवस्थित और संरचित बनाने के लिए एक स्वचालित प्रक्रिया जैसे बुद्धिमान ऐप्स के साथ संयुक्त होने पर, उद्योगों को आसानी से डेटा का अध्ययन और मिलान करने, जानकारी का मूल्यांकन करने और स्केलेबिलिटी की योजना बनाने में मदद मिलती है। नतीजतन, जटिल व्यावसायिक समस्याओं के अधिक व्यवहार्य समाधान सामने आते हैं।
4. वायरस और अन्य समझौता करने वाले सॉफ्टवेयर से सुरक्षा - आईटी समर्थन सेवाएं सुनिश्चित करती हैं कि कंप्यूटर सिस्टम विभिन्न प्रकार के वायरस और अन्य इंटरनेट खतरों से सुरक्षित हैं। आपके उपकरणों के लिए व्यापक सुरक्षा प्रदान करने के लिए मानक एंटीवायरस प्रबंधन के संयोजन को स्थापित करना आईटी विभाग का काम है। अगर वे अपने कंप्यूटर सिस्टम को अपडेट और अच्छी तरह से मॉनिटर करते हैं तो फर्म डिजिटल डेटा एक्सेस और संचालन के खतरों के शिकार होने से बच जाएगी।
5. व्यापक निगरानी - हर स्तर पर, कंपनी के आंतरिक संचालन और उपभोक्ता आउटरीच गतिविधियों के प्रदर्शन और प्रगति को ट्रैक करना महत्वपूर्ण है। विनिर्माण स्थानों वाले संगठनों के लिए गुणवत्ता नियंत्रण, सुविधाओं की योजना और रसद में सुधार, और आंतरिक ऑडिट कुछ ऐसे बेहतर तरीके हैं जिनसे आईटी व्यवसाय के आवश्यक कार्यों के अधिक परिष्कृत पर्यवेक्षण को निष्पादित करने में मदद कर सकता है। ग्राहकों को ऑनलाइन सेवाएं प्रदान करने वाली फर्मों के लिए एक सक्षम आईटी प्रणाली की मदद से व्यापक निगरानी भी एक आवश्यकता है। यह सुनिश्चित करने के लिए है कि उनकी सेवाओं के साथ-साथ उनके ग्राहकों की सुरक्षा को कोई खतरा न हो।
6. कंपनी जनशक्ति और मानव संसाधन प्रबंधन को व्यवस्थित करें - अधिक हार्ड-टेक और प्रबंधित रिकॉर्ड-कीपिंग विकल्पों की उपलब्धता को देखते हुए पेपर-आधारित पेपर अब केवल कुशल या व्यावहारिक नहीं हैं। एक सूचना प्रणाली को कंपनी की विशिष्ट संरचना और काम पर रखने की प्रथाओं के अनुरूप बनाया जा सकता है, यह एक और उदाहरण प्रदान करता है कि स्टार्ट-अप और तेजी से बढ़ती फर्मों के लिए आईटी समर्थन क्यों महत्वपूर्ण है।
7. उन्नत ऑनलाइन मार्केटिंग रणनीतियाँ - सूचना प्रणाली अधिक सटीक बाजार अनुसंधान की सुविधा और उपयोगी डेटा एकत्र करके विपणन रणनीति में मदद कर सकती है। लक्षित दर्शकों को ढूँढना, उनकी विशेष आवश्यकताओं और इच्छाओं का निर्धारण करना, और एक प्रचार रणनीति बनाना जो उन्हें खरीदने के लिए लुभाती है, ये सभी इस प्रक्रिया का हिस्सा हैं।
8. बेहतर ग्राहक सहायता - ग्राहक सहायता प्रणालियों में प्रौद्योगिकी को लागू करने के लिए आउटसोर्सिंग आईटी समर्थन के लाभों का उपयोग करना एक अन्य विकल्प है। स्टार्ट-अप के पास एक छोटा कार्यबल है, जिससे प्रश्नों और ग्राहकों की शिकायतों की बढ़ती मात्रा को बनाए रखना मुश्किल हो जाता है क्योंकि उनकी सेवाओं और दर्शकों तक पहुंच विकसित होती है। कंपनी के समर्थन एजेंटों के मौजूदा कार्यबल को बढ़ाने के लिए दूरस्थ कर्मियों को काम पर रखना एक विश्वसनीय आईटी प्रणाली के साथ संभव है। किसी भी प्रकार के व्यवसाय के लिए आईटी सहायता

सेवाएं आवश्यक हैं, चाहे वह एक स्टार्ट-अप हो या एक अच्छी तरह से स्थापित उद्यम। यह न केवल सिस्टम को बनाए रखने के लिए महत्वपूर्ण है, बल्कि निरंतर अपडेट के माध्यम से उत्कृष्टता प्राप्त करने के लिए भी है जो आपकी कंपनी के संचालन के इष्टतम स्तर को सुनिश्चित कर सकता है।

9. बेहतर ब्रांडिंग को बढ़ावा दें - ब्रांडिंग पहल विकसित करने पर तकनीकी सहायता का जो प्रभाव है, वह इस बात का अंतिम लेकिन कम से कम उदाहरण नहीं है कि तकनीकी सहायता महत्वपूर्ण क्यों है। जब ब्रांडिंग को सूचना सेवाओं और प्रणालियों के साथ जोड़ा जाता है, तो इसका उपयोग मौजूदा मार्केटिंग रणनीति में सुधार के साथ-साथ नए बनाने में मदद के लिए किया जा सकता है। आईटी कंपनी के उत्पाद और सेवा प्रसाद की विशिष्टता पर जोर देकर ब्रांडिंग बढ़ाने में मदद कर सकता है।

### 2.1.2 सॉफ्टवेयर डेवलपमेंट में अनुप्रयोग (एप्लीकेशन)

सॉफ्टवेयर डेवलपर्स संपूर्ण सॉफ्टवेयर डेवलपमेंट प्रक्रिया के प्रभारी हैं। वे ग्राहक से पूछकर शुरू कर सकते हैं कि वे सॉफ्टवेयर का उपयोग कैसे करना चाहते हैं। उन्हें मूल कार्यक्षमता की पहचान करनी चाहिए जो सॉफ्टवेयर उपयोगकर्ताओं को चाहिए। सॉफ्टवेयर डेवलपर्स को सॉफ्टवेयर के कार्यों से असंबंधित उपयोगकर्ता आवश्यकताओं को भी निर्धारित करना चाहिए, जैसे सुरक्षा और प्रदर्शन आवश्यकताओं का स्तर। वे प्रोग्राम को डिज़ाइन करते हैं और फिर प्रोग्रामर्स को निर्देश देते हैं, जो कंप्यूटर कोड लिखते और उसका परीक्षण करते हैं।

यदि प्रोग्राम अपेक्षित रूप से काम नहीं करता है या यदि परीक्षकों को इसका उपयोग करना बहुत मुश्किल लगता है, तो सॉफ्टवेयर डेवलपर्स, प्रोग्राम को ठीक करने या सुधारने के लिए डिज़ाइन प्रक्रिया पर वापस आ जाते हैं। ग्राहक को जारी किए जाने के बाद एक डेवलपर प्रोग्राम पर अपग्रेड और रखरखाव कर सकता है। डेवलपर्स आमतौर पर कंप्यूटर प्रोग्रामर के साथ मिलकर काम करते हैं।

हालाँकि, कुछ कंपनियों में, डेवलपर्स प्रोग्रामर को निर्देश देने के बजाय कोड लिखते हैं। डेवलपर्स जो एक सॉफ्टवेयर प्रोजेक्ट की योजना बनाने से लेकर कार्यान्वयन तक की देखरेख करते हैं, उन्हें कभी-कभी सूचना प्रौद्योगिकी (आईटी) परियोजना प्रबंधकों के रूप में जाना जाता है। ये कर्मचारी यह सुनिश्चित करने के लिए परियोजना की प्रगति पर नज़र रखते हैं कि यह समय सीमा, मानकों और बजटीय बाधाओं को पूरा करता है। कंप्यूटर और सूचना प्रणाली प्रबंधकों के प्रोफाइल में आईटी परियोजना प्रबंधक शामिल हैं जो किसी संगठन के आईटी विभाग या आईटी नीतियों की योजना बनाते हैं और उन्हें निर्देशित करते हैं। सॉफ्टवेयर डेवलपर्स के प्रकारों के उदाहरण निम्नलिखित हैं:

एप्लिकेशन सॉफ्टवेयर डेवलपर उपभोक्ताओं के लिए कंप्यूटर एप्लिकेशन बनाते हैं, जैसे वर्ड प्रोसेसर और गेम। वे किसी विशिष्ट ग्राहक के लिए कस्टम सॉफ्टवेयर या सामान्य बिक्री के लिए व्यावसायिक सॉफ्टवेयर विकसित कर सकते हैं। कुछ एप्लिकेशन सॉफ्टवेयर डेवलपर जटिल डेटाबेस डिज़ाइन करते हैं। वे इंटरनेट पर और कंपनी के इंटरनेट के भीतर उपयोग के लिए प्रोग्राम भी विकसित करते हैं।

सिस्टम सॉफ्टवेयर डेवलपर उन सिस्टमों को डिज़ाइन करते हैं जो कंप्यूटर को चालू रखते हैं। ये आम जनता द्वारा खरीदे गए कंप्यूटरों के लिए ऑपरेटिंग सिस्टम या विशेष रूप से किसी कंपनी के लिए बनाए गए सिस्टम हो सकते हैं। सिस्टम सॉफ्टवेयर डेवलपर अक्सर सिस्टम का इंटरफ़ेस बनाते हैं, जो उपयोगकर्ताओं को कंप्यूटर के साथ इंटरैक्ट करने की अनुमति देता है। सिस्टम सॉफ्टवेयर डेवलपर ऑपरेटिंग सिस्टम बनाते हैं जो सेल फोन और ऑटोमोबाइल सहित आज के अधिकांश उपभोक्ता इलेक्ट्रॉनिक्स को शक्ति प्रदान करते हैं। सॉफ्टवेयर विकास ऐसे एप्लिकेशन बनाता है जो व्यवसायों को डिजिटल दुनिया में प्रतिस्पर्धा करने में सहायता करते हैं। आईबीएम के अनुसार, "आधे से अधिक कंपनियां यह मानती हैं कि प्रतिस्पर्धात्मक लाभ प्राप्त करने के लिए विचारधारा से वितरण तक प्रभावी सॉफ्टवेयर विकास महत्वपूर्ण है।

सॉफ्टवेयर डेवलपमेंट व्यवसायों के लिए मूल्य जोड़ता है और उन्हें एप्लिकेशन प्रकारों में प्रतिस्पर्धा करने में मदद करता है:



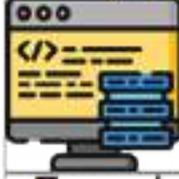
वेब इंटरनेट एप्लिकेशन जो ईमेल से लेकर खुदरा लेनदेन तक हर चीज का समर्थन करते हैं, लॉग और वेबनर उत्पाद और सेवाएं प्रदान करते हैं और बाईं को असंग करते हैं।



वेब इंटरनेट एप्लिकेशन जो ईमेल से लेकर खुदरा लेनदेन तक हर चीज का समर्थन करते हैं, लॉग और वेबनर उत्पाद और सेवाएं प्रदान करते हैं और बाईं को असंग करते हैं।



वेब इंटरनेट एप्लिकेशन जो ईमेल से लेकर खुदरा लेनदेन तक हर चीज का समर्थन करते हैं, लॉग और वेबनर उत्पाद और सेवाएं प्रदान करते हैं और बाईं को असंग करते हैं।



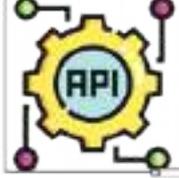
Microsoft Windows, Apple Mac OS, और Linux जैसे ऑपरेटिंग सिस्टम पर चलने वाले व्यावसायिक अनुप्रयोग कोर ऑपरेटिंग एप्लिकेशन उत्पादकता बढ़ाते हैं और संचालन को सुव्यवस्थित करते हैं।



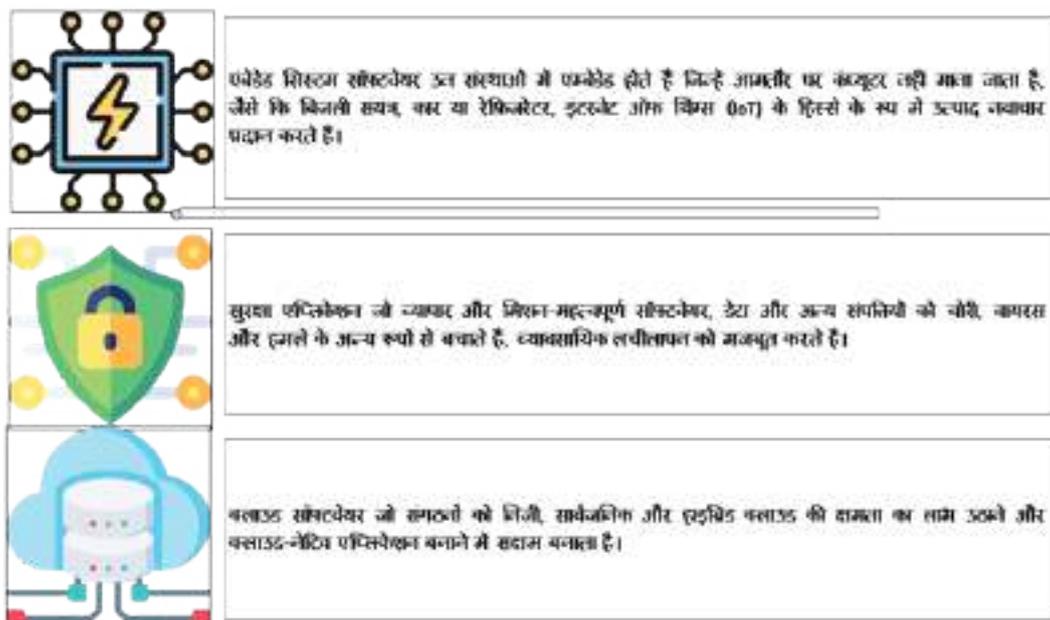
बैक-एंड डेवलपमेंट इंटीग्रेशन, डेटाबेस और मॉडलिंग क्लिंट-फेसिंग एप्लिकेशन को सुचारु रूप से चलाने, गहरी जानकारी देने और क्लिंट एक्सपीरियंस को समृद्ध करने में सक्षम बनाती हैं।



सॉफ्टवेयर उपकरण अनुप्रयोग विकास की गति, प्रक्रियाओं और विश्वसनीयता को बढ़ाते हैं और अनुप्रयोग जीवनचक्र प्रबंधन में सुधार करते हैं।



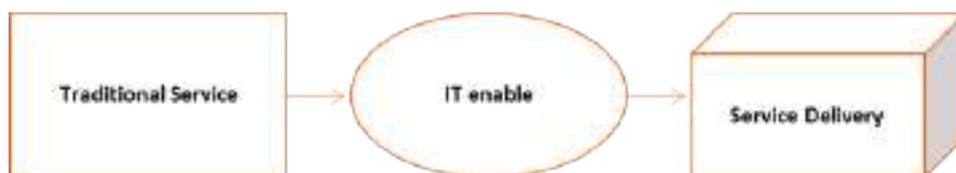
एपीआई एप्लिकेशन प्रोवाइंग इंटरनेट सॉफ्टवेयर प्रोवाइंग को एक दूसरे से जुड़ने और बातचीत करने और लॉग व्यावसायिक एप्लिकेशंस और अद्यतन बनाने में सक्षम बनाता है।



चित्र 2.2 सॉफ्टवेयर विकास में अनुप्रयोगों के प्रकार

### 2.1.3 आईटी सक्षम सेवाएं

आईटी-सक्षम सेवाएं वे हैं जो सेवा की गुणवत्ता में सुधार करके व्यवसायों की सहायता के लिए सूचना प्रौद्योगिकी का उपयोग करती हैं। आईटीईएस "आईटी सक्षम सेवाओं" के लिए एक संक्षिप्त शब्द है। एक आईटी-सक्षम सेवा में निम्नलिखित बुनियादी संरचना होती है:



चित्र 2.3 कारोबारी प्रक्रिया की पुनर्रचना

(Source: [https://cio-wiki.org/wiki/Information\\_Technology\\_Enabled\\_Services\\_\(ITeS\)\)](https://cio-wiki.org/wiki/Information_Technology_Enabled_Services_(ITeS)))

सबसे महत्वपूर्ण कारक आईटी-सक्षम सेवाओं का अतिरिक्त मूल्य है। ग्राहक संबंध प्रबंधन, बेहतर डेटाबेस, बेहतर रूप और अनुभव, और इसी तरह मूल्यवर्धन के उदाहरण हैं। एक आईटी-सक्षम सेवा दो प्रकार के परिणाम उत्पन्न करती है:

प्रत्यक्ष बेहतर सेवा

अप्रत्यक्ष लाभ

जबकि प्रत्यक्ष लाभ तुरंत प्राप्त होते हैं, अप्रत्यक्ष लाभ समय के साथ अर्जित होते हैं और यदि समय से पहले अच्छी तरह से योजना बनाई जाए तो इसका बहुत प्रभावी ढंग से उपयोग किया जा सकता है। संचालन की पूरी श्रृंखला जो किसी संगठन की दक्षता में सुधार के लिए सूचना प्रौद्योगिकी का उपयोग करती है।

ये सेवाएं कॉल सेंटर, मेडिकल ट्रांसक्रिप्शन, मेडिकल बिलिंग और कोडिंग, बैंक-ऑफिस संचालन में रोजगार के अवसर प्रदान करती हैं; राजस्व दावा प्रसंस्करण, कानूनी डेटाबेस, सामग्री विकास, पेट्रोल, रसद प्रबंधन, जीआईएस (भौगोलिक सूचना प्रणाली), मानव संसाधन सेवाएं, वेब सेवाएं और अन्य क्षेत्र।

## 2.1.4 आईटी-सक्षम सेवाओं की कुंजी

सबसे महत्वपूर्ण आईटी-सक्षम सेवाएं निम्नलिखित हैं:



चित्र 2.4 सक्षम सेवाओं की कुंजी

दी जाने वाली कुछ सेवाओं में निम्न शामिल हैं:



चित्र 2.5 दी जाने वाली सेवाएँ

### 2.1.5 अवसर और चुनौतियाँ

आर्थिक और व्यावसायिक परिस्थितियों में बदलाव, तेजी से तकनीकी नवाचार, इंटरनेट प्रसार और वैश्वीकरण सभी तेजी से प्रतिस्पर्धी माहौल में योगदान दे रहे हैं। प्रौद्योगिकी की भूमिका समर्थन से बदलकर निगमों को बदलने में स्थानांतरित हो गई है। उच्च गुणवत्ता, कम लागत वाले प्रौद्योगिकी समाधानों की मांग को पूरा करने के लिए वैश्विक निगम तेजी से अपतटीय प्रौद्योगिकी सेवा प्रदाताओं की ओर रुख कर रहे हैं। दुनिया भर के लगभग 80 देशों में भारतीय आईटी-सक्षम सेवा कंपनियों द्वारा एक हजार से अधिक वैश्विक वितरण केंद्र स्थापित किए गए हैं। भारत की आईटी-सक्षम सेवाओं के पीछे मौजूद अवसर इस प्रकार हैं:



चित्र 2.6 आईटी/आईटीईएस सक्षम सेवाओं में अवसर

आईटी-सक्षम सेवाएं विभिन्न प्रकार की आईटी प्रक्रियाओं और सेवाओं की पेशकश करती हैं। जबकि आईटी क्षेत्र की विकास संभावनाएं आशाजनक हैं, आईटी-सक्षम सेवाओं की स्थापना करते समय कई कार्यान्वयन चुनौतियों का सामना करना पड़ता है। यह समझ में आता है कि आईटी और आईटीईएस उद्योग में एक कंपनी के रूप में, वे अपने प्रयासों में जोखिमों और चुनौतियों की एक विस्तृत श्रृंखला का सामना करते हैं। एक निर्बाध, सफल, टिकाऊ और स्केलेबल व्यवसाय बनाने और बनाए रखने के लिए। सामने आने वाली कठिनाइयों में से हैं:



चित्र 2.7 आईटी/आईटीईएस सक्षम सेवाओं में चुनौतियाँ

## 2.1.6 शामिल प्रौद्योगिकियां

प्रत्येक सेवा में दो प्रकार की प्रौद्योगिकियां शामिल होती हैं:

तकनीक सक्षम करना

संचार

सक्षम करने वाली प्रौद्योगिकियां वे हैं जो आईटी सक्षम करने की अनुमति देती हैं। कॉल सेंटर्स में, उदाहरण के लिए, वे सीटीआई (कंप्यूटर टेलीफोनी इंटीग्रेशन), सीआरएम (ग्राहक संबंध प्रबंधन) आदि हैं। संचार आईटी सेवाओं को एक नए चैनल-टेलीफोन, आईपी, वीओआईपी (वॉयस ओवर आईपी: वॉयस ओवर इंटरनेट प्रोटोकॉल), सैटेलाइट ट्रांसमिशन, आदि के माध्यम से वितरित करने में सक्षम बनाता है। जब सही तकनीक और संचार चैनल संयुक्त होते हैं, तो सेवा की गुणवत्ता में नाटकीय रूप से सुधार होता है।

## 2.1.7 परिनियोजन मुद्दे

एक परिनियोजन समस्या तब होती है जब फ़ाइलों का गलत सेट साइट फ़ोल्डर (आमतौर पर d: होम/साइट/www-रूट) पर तैनात किया जाता है, या जब कुछ फ़ाइलें बिल्कुल भी तैनात नहीं होती हैं। परिनियोजन के परिणामस्वरूप फ़ाइलों के एक निश्चित सेट के उस फ़ोल्डर में समाप्त होने की अपेक्षा करें, लेकिन परिनियोजन के दौरान विफलता के परिणामस्वरूप फ़ाइलों का एक अलग सेट होता है।

अनुप्रयोग परिनियोजन विफलता के छह कारण निम्न हैं -

प्रक्रिया की अपर्याप्तता - परिचालन लचीलापन असफलताओं से उबरने की क्षमता से कहीं अधिक आवश्यक है। इसमें निवारक उपायों के माध्यम से विफलता से बचने की क्षमता भी शामिल है। कई व्यवसायों और सूचना प्रौद्योगिकी विभागों में सफलता के लिए आवश्यक परिचालन लचीलापन परिपक्वता की कमी है। हालांकि एप्लिकेशन विफलताओं को रोकना लगभग असंभव है, संगठनों को उन्हें पहचानने, भविष्यवाणी करने और उन्हें ठीक करने का प्रयास करना चाहिए।

रिलीज पाइपलाइन असंगत है - कई संगठनों के पास अपने आईटी सिस्टम के माध्यम से सॉफ्टवेयर परिनियोजन मॉडल का गलत संरेखण है। चूंकि आईटी परिदृश्य में सिस्टम आम तौर पर एक दूसरे से जुड़े होते हैं, परिणामस्वरूप विफलताएं होती हैं।

प्रक्रिया जटिलताएं - विकास और परिचालन टीमों द्वारा उपयोग किए जाने वाले टूलसेट और परिनियोजन प्रक्रियाओं की विविधता कुछ वातावरण को जटिल बनाती है। बड़ी संख्या में टूल उनके बीच एम्बेडेड मैनुअल प्रक्रियाओं के साथ कई टूलिंग डोमेन बनाते हैं, जिसके परिणामस्वरूप प्रक्रिया जटिलता होती है। इसके अलावा, ऐसे समय होते हैं जब रिलीज पाइपलाइन के विपरीत छोर पर प्रावधान और परिनियोजन प्रक्रियाओं का विरोध किया जाता है।

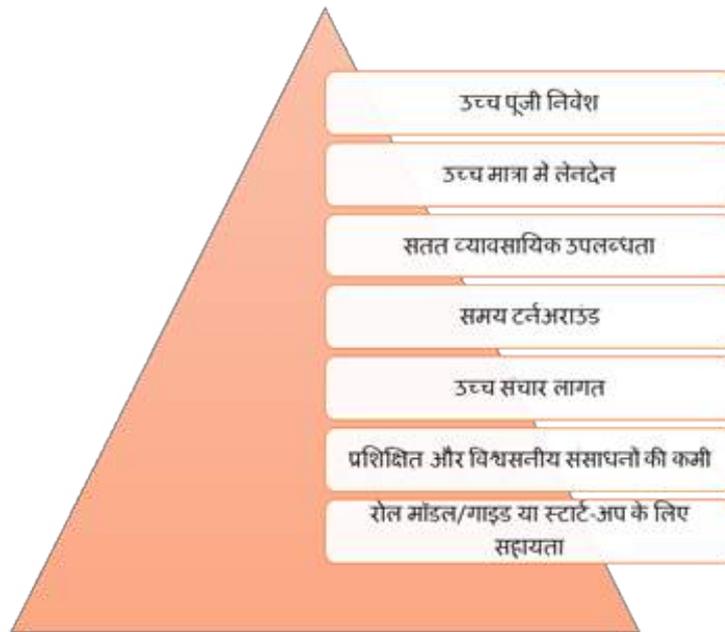
अवज्ञा - विकास और रिलीज प्रक्रिया के दौरान, एप्लिकेशन भेद्यता स्कैनिंग अक्सर मानकीकरण और लचीलेपन की कमी को प्रकट करती है। उचित शासन उपायों की कमी के कारण, विकास दल उचित सुरक्षा परीक्षण करने में विफल होते हैं, जिसके परिणामस्वरूप ये खामियां होती हैं।

कुछ मामलों में, परीक्षण को महंगा और समय लेने वाला माना जाता है, जिससे प्रयास को कम करने की प्रवृत्ति होती है।

क्षमताओं की कमी - प्रत्येक संगठन का अपना नायक डेवलपर या संचालन विशेषज्ञ होता है जो किसी भी समस्या को अपने दम पर हल करने में सक्षम होता है। समय के साथ, इन व्यक्तियों के इर्द-गिर्द प्रक्रियाएं बन जाती हैं, जिससे उनके जाने पर उन्हें चलाना मुश्किल हो जाता है। ऐसी प्रक्रियाओं का होना महत्वपूर्ण है जो न केवल एक या दो महत्वपूर्ण संसाधनों के आसपास बनाई गई हैं बल्कि संगठन की बदलती मांगों को पूरा करने के लिए स्केलेबल, दोहराने योग्य और स्वचालित भी हैं।

अनिश्चितता - आईटी, विकास और परिचालन टीमों की मांग और आपूर्ति पक्षों के बीच उचित संचार और अंतःक्रियाशीलता की कमी के परिणामस्वरूप ऐसी परिस्थितियां उत्पन्न होती हैं जहां कार्रवाई अलगाव में की जाती है और तार्किक दिखाई देती है लेकिन संयुक्त अंत तक संयुक्त रूप से विफल हो जाती है। कई संगठनों में, अधिकांश परिवर्तन मामूली परिवर्धन या परिवर्तन होते हैं। प्रमुख परियोजनाओं के विपरीत, ये परिवर्तन अक्सर कम निरीक्षण और नियंत्रण के अधीन होते हैं।

जबकि विकास मजबूत है और संभावनाएं उज्ज्वल हैं, आईटी-सक्षम सेवाओं की स्थापना में कई तैनाती चुनौतियां हैं।



चित्र 2.8 तैनाती के मुद्दे

## सारांश

- सूचना प्रौद्योगिकी वाणिज्य और व्यावसायिक क्षेत्रों के विकास और विस्तार में सहायता करती है, जिसके परिणामस्वरूप अधिकतम उत्पादन होता है। सूचना प्रौद्योगिकी में प्रगति के कारण विभिन्न क्षेत्रों को व्यवसाय उत्पन्न करने में लगने वाला समय कम हो गया है। यह इलेक्ट्रॉनिक सुरक्षा, डेटा का भंडारण और प्रभावी संचार प्रदान करता है।
- सॉफ्टवेयर डेवलपर्स संपूर्ण सॉफ्टवेयर विकास प्रक्रिया के प्रभारी हैं। वे ग्राहक से पूछकर शुरू कर सकते हैं कि वे सॉफ्टवेयर का उपयोग कैसे करना चाहते हैं। उन्हें मूल कार्यक्षमता की पहचान करनी चाहिए जो सॉफ्टवेयर उपयोगकर्ताओं को चाहिए। सॉफ्टवेयर डेवलपर्स को सॉफ्टवेयर के कार्यों से असंबंधित उपयोगकर्ता आवश्यकताओं को भी निर्धारित करना चाहिए, जैसे सुरक्षा और प्रदर्शन आवश्यकताओं का स्तर।
- एप्लिकेशन सॉफ्टवेयर डेवलपर उपभोक्ताओं के लिए कंप्यूटर एप्लिकेशन बनाते हैं, जैसे वर्ड प्रोसेसर और गेम। वे किसी विशिष्ट ग्राहक के लिए कस्टम सॉफ्टवेयर या सामान्य बिक्री के लिए व्यावसायिक सॉफ्टवेयर विकसित कर सकते हैं। कुछ एप्लिकेशन सॉफ्टवेयर डेवलपर जटिल डेटाबेस डिज़ाइन करते हैं।
- सिस्टम सॉफ्टवेयर डेवलपर उन सिस्टमों को डिज़ाइन करते हैं जो कंप्यूटर को चालू रखते हैं। ये आम जनता द्वारा खरीदे गए कंप्यूटरों के लिए ऑपरेटिंग सिस्टम या विशेष रूप से किसी कंपनी के लिए बनाए गए सिस्टम हो सकते हैं।
- सॉफ्टवेयर उपकरण अनुप्रयोग विकास की गति, प्रक्रियाओं और विश्वसनीयता को बढ़ाते हैं और अनुप्रयोग जीवनचक्र प्रबंधन में सुधार करते हैं।
- सबसे महत्वपूर्ण कारक आईटी-सक्षम सेवाओं का अतिरिक्त मूल्य है। ग्राहक संबंध प्रबंधन, बेहतर डेटाबेस, बेहतर रूप और अनुभव, और इसी तरह मूल्यवर्धन के उदाहरण हैं।
- आईटी सक्षम सेवाएं कॉल सेंटर, मेडिकल ट्रांसक्रिप्शन, मेडिकल बिलिंग और कोडिंग, बैंक-ऑफिस संचालन में रोजगार के अवसर प्रदान करती हैं; राजस्व दावा प्रसंस्करण, कानूनी डेटाबेस, सामग्री विकास, पेट्रोल, रसद प्रबंधन, जीआईएस (भौगोलिक सूचना प्रणाली), मानव संसाधन सेवाएं, वेब सेवाएं और अन्य क्षेत्र।
- आर्थिक और व्यावसायिक परिस्थितियों में बदलाव, तेजी से तकनीकी नवाचार, इंटरनेट प्रसार और वैश्वीकरण सभी तेजी से प्रतिस्पर्धी माहौल में योगदान दे रहे हैं। प्रौद्योगिकी की भूमिका समर्थन से बदलकर निगमों को बदलने में स्थानांतरित हो गई है। उच्च गुणवत्ता, कम लागत वाले प्रौद्योगिकी समाधानों की मांग को पूरा करने के लिए वैश्विक निगम तेजी से अपतटीय प्रौद्योगिकी सेवा प्रदाताओं की ओर रुख कर रहे हैं।
- आईटी-सक्षम सेवाएं विभिन्न प्रकार की आईटी प्रक्रियाओं और सेवाओं की पेशकश करती हैं। जबकि आईटी क्षेत्र की विकास संभावनाएं आशाजनक हैं, आईटी-सक्षम सेवाओं की स्थापना करते समय कई कार्यान्वयन चुनौतियों का सामना करना पड़ता है।
- सक्षम करने वाली प्रौद्योगिकियां वे हैं जो आईटी सक्षम करने की अनुमति देती हैं। कॉल सेंटरों में, उदाहरण के लिए, वे सीटीआई (कंप्यूटर टेलीफोनी इंटीग्रेशन), सीआरएम (ग्राहक संबंध प्रबंधन) आदि हैं। संचार आईटी सेवाओं को एक नए चैनल-टेलीफोन, आईपी, वीओआईपी (वॉयस ओवर आईपी: वॉयस ओवर इंटरनेट प्रोटोकॉल), सैटेलाइट ट्रांसमिशन, आदि के माध्यम से वितरित करने में सक्षम बनाता है। जब सही तकनीक और संचार चैनल संयुक्त होते हैं, तो सेवा की गुणवत्ता में नाटकीय रूप से सुधार होता है।

## अभ्यास



I. रिक्त स्थान की पूर्ती करो

- a) \_\_\_\_\_ अनुप्रयोग विकास की गति, प्रक्रियाओं और विश्वसनीयता को बढ़ाता है और अनुप्रयोग जीवनचक्र प्रबंधन में सुधार करता है।
- b) \_\_\_\_\_ वाणिज्य और व्यावसायिक क्षेत्रों के विकास और विस्तार में सहायता करता है, जिसके परिणामस्वरूप अधिकतम उत्पादन होता है।
- c) \_\_\_\_\_ उपभोक्ताओं के लिए वर्ड प्रोसेसर और गेम जैसे कंप्यूटर एप्लिकेशन बनाते हैं।
- d) \_\_\_\_\_ आईटी सेवाओं को एक नए चैनल-टेलीफोन, आईपी, वीओआईपी, सैटेलाइट ट्रांसमिशन आदि के माध्यम से वितरित करने में सक्षम बनाता है।
- e) \_\_\_\_\_ कॉल सेंटर, मेडिकल ट्रांसक्रिप्शन, मेडिकल बिलिंग और कोडिंग, बैंक-ऑफिस संचालन में रोजगार के अवसर प्रदान करते हैं।

II. निम्नलिखित प्रश्नों के उत्तर दीजिए

1. आईटी उद्योग में तैनाती (डेप्लॉयमेंट) के मुद्दे क्या हैं?
2. आईटी क्षेत्र में कितनी प्रौद्योगिकियां शामिल हैं?
3. आईटी-सक्षम सेवाओं की कुंजी समझाइए।
4. सॉफ्टवेयर विकास के अनुप्रयोगों के प्रकारों की व्याख्या कीजिए।
5. आईटी क्षेत्र के अवसरों और चुनौतियों की व्याख्या करें।



## 3. सॉफ्टवेयर डेवलपमेंट की प्रक्रिया



IT - ITeS SSC  
NASSCOM

यूनिट 3.1 सॉफ्टवेयर डेवलपमेंट जीवनचक्र



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. आवश्यकता के अनुसार सॉफ्टवेयर को डिजाइन, परीक्षण और विकसित करने के लिए उपयोगकर्ताओं की जरूरतों का विश्लेषण करें
2. सॉफ्टवेयर डेवलपमेंट प्रक्रिया को सुविधाजनक बनाने के लिए मॉडल और दृष्टिकोण डिजाइन करने के लिए विभिन्न चरणों का मूल्यांकन करें

## यूनिट 3.1: सॉफ्टवेयर डेवलपमेंट प्रक्रिया

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. सॉफ्टवेयर डेवलपमेंट जीवनचक्र के चरणों की सूची बनाएं
2. टॉप-डाउन और बॉटम-अप डिज़ाइन दृष्टिकोणों के बीच अंतर पर चर्चा करें

### 3.1.1 सॉफ्टवेयर डेवलपमेंट प्रक्रिया

एक प्रक्रिया, तब, एक विशिष्ट क्रम में की जाने वाली क्रियाओं की एक श्रृंखला है। एक सॉफ्टवेयर डेवलपमेंट प्रक्रिया उन चरणों से बनी होती है जो एक तैयार उत्पाद की ओर ले जाते हैं। सॉफ्टवेयर डेवलपमेंट प्रक्रिया चरणों की एक श्रृंखला है जिसे डिलिवरेबल्स का उत्पादन करने के लिए पूरा किया जाना चाहिए। सॉफ्टवेयर डेवलपमेंट प्रक्रिया, एसडीएलसी के विपरीत, प्रत्येक चरण को निर्दिष्ट करती है जिसे डिलिवरेबल्स के हिस्से के रूप में पूरा किया जाना चाहिए जिसमें प्रत्येक चरण शामिल है। यह छोटे पैमाने पर चरणों की एक श्रृंखला है।

### 3.1.2 एसडीएलसी (SDLC) चरण

<p>आवश्यकता सभा और विश्लेषण</p> 	<p>इस चरण के दौरान, ग्राहक से सभी प्रासंगिक जानकारी एक उत्पाद विकसित करने के लिए एकत्र की जाती है जो उनकी अपेक्षाओं को पूरा करता है।</p> <p>इस चरण के दौरान ही किसी भी अस्पष्टता का समाधान किया जाना चाहिए।</p> <p>व्यापार विश्लेषक और परियोजना प्रबंधक सभी आवश्यक जानकारी एकत्र करने के लिए ग्राहक के साथ एक बैठक का समय निर्धारित करते हैं, जैसे कि ग्राहक क्या बनाना चाहता है, अंतिम उपयोगकर्ता कौन होगा, और उत्पाद का उद्देश्य क्या है।</p> <p>किसी उत्पाद को विकसित करने से पहले, उत्पाद की ठोस समझ होना महत्वपूर्ण है।</p>
<p>डिजाईन</p> 	<p>इस चरण में, एसआरएस दस्तावेज़ में एकत्रित आवश्यकता को एक इनपुट के रूप में उपयोग किया जाता है, और सिस्टम विकास को लागू करने के लिए उपयोग किया जाने वाला सॉफ्टवेयर आर्किटेक्चर प्राप्त होता है।</p>

<p>कार्यान्वयन या कोडिंग</p> 	<p>एक बार डेवलपर को डिज़ाइन दस्तावेज़ मिल जाता है तो कार्यान्वयन/कोडिंग शुरू हो जाती है।</p> <p>सॉफ्टवेयर डिज़ाइन का स्रोत कोड में अनुवाद किया गया है। इस चरण में सॉफ्टवेयर के सभी घटकों को लागू किया जाता है।</p>
<p>परीक्षण</p> 	<p>कोडिंग पूर्ण होने के बाद परीक्षण शुरू होता है और परीक्षण के लिए माँड्यूल जारी किए जाते हैं। इस चरण में, विकसित सॉफ्टवेयर का पूरी तरह से परीक्षण किया जाता है और जो भी दोष पाए जाते हैं उन्हें ठीक करने के लिए डेवलपर्स को सौंपा जाता है।</p> <p>पुनः परीक्षण, और प्रतिगमन परीक्षण उस बिंदु तक किया जाता है जिस पर सॉफ्टवेयर ग्राहक की अपेक्षा के अनुसार होता है। परीक्षक यह सुनिश्चित करने के लिए एसआरएस दस्तावेज़ देखें कि सॉफ्टवेयर ग्राहक के मानक के अनुसार है।</p>
<p>डेप्लॉयमेंट</p> 	<p>एक बार उत्पाद का परीक्षण हो जाने के बाद, इसे या तो उत्पादन में तैनात किया जाता है या ग्राहक की अपेक्षाओं के आधार पर पहला यूएटी (उपयोगकर्ता स्वीकृति परीक्षण) किया जाता है।</p> <p>यूएटी के मामले में, उत्पादन वातावरण की एक प्रतिकृति बनाई जाती है, और ग्राहक डेवलपर्स के साथ परीक्षण करता है।</p> <p>यदि ग्राहक एप्लिकेशन को अपेक्षा के अनुरूप पाता है, तो ग्राहक एप्लिकेशन के लाइव होने पर हस्ताक्षर कर देता है।</p>
<p>रखरखाव</p> 	<p>उत्पादन वातावरण में उत्पाद की तैनाती के बाद, डेवलपर्स उत्पाद के रखरखाव के लिए जिम्मेदार होते हैं, यानी, यदि कोई समस्या उत्पन्न होती है जिसे हल करने की आवश्यकता होती है या एक वृद्धि की आवश्यकता होती है।</p>

सूची 3.1 SDLC के चरण

### 3.1.3 टॉप-डाउन और बॉटम-अप डिजाइन दृष्टिकोण

टॉप-डाउन डिजाइन दृष्टिकोण

टॉप-डाउन प्रोग्रामिंग बॉटम-अप प्रोग्रामिंग का विलोम है। यह एक प्रोग्रामिंग शैली को संदर्भित करता है जहां एक एप्लिकेशन को उच्च-स्तरीय विवरण से बनाया जाता है कि उसे क्या करना चाहिए और तब तक सरल और सरल टुकड़ों में टूट जाता है जब तक कि एक स्तर तक नहीं पहुंच जाता है जो प्रोग्रामिंग भाषा के प्राइमेटिव से मेल खाती है। .



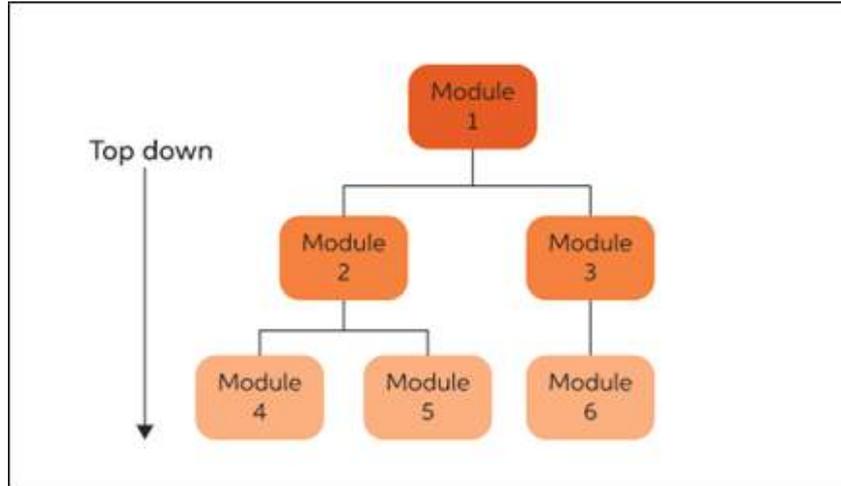
चित्र 3.1 टॉप-डाउन प्रोग्रामिंग के लाभ



चित्र . 3.2 टॉप-डाउन प्रोग्रामिंग के नुकसान

टॉप-डाउन प्रोग्रामिंग पर काम करना

टॉप-डाउन प्रोग्रामिंग कार्यक्षमता के आधार पर मॉड्यूल का उत्पादन करती है, आमतौर पर कार्यों या प्रक्रियाओं के रूप में। आमतौर पर, सिस्टम का उच्च-स्तरीय विनिर्देश कार्यक्षमता बताता है। यह उच्च-स्तरीय विवरण सरल कार्यों या प्रक्रियाओं के एक त्रूप में परिष्कृत हो जाता है और बाद में इसे और परिष्कृत किया जाता है। इस प्रोग्रामिंग शैली में एक उच्च जोखिम है कि कई डेटा संरचनाओं के कार्यान्वयन विवरण को मॉड्यूल के बीच साझा किया जाना चाहिए और इस प्रकार विश्व स्तर पर उजागर किया जाना चाहिए। यह अन्य मॉड्यूल को इन कार्यान्वयन विवरणों का उपयोग करने के लिए प्रोत्साहित करता है, जिसके परिणामस्वरूप एप्लिकेशन के विभिन्न हिस्सों के बीच अवांछित निर्भरता होती है।



चित्र 3.3 टॉप डाउन डिजाईन

(Image Source: <https://www.chegg.com/learn/computer-science/computer-software/top-down-design>)

### बॉटम-अप प्रोग्रामिंग

बॉटम-अप प्रोग्रामिंग टॉप-डाउन प्रोग्रामिंग का विलोम है। यह एक प्रोग्रामिंग शैली को संदर्भित करता है जिसमें एक एप्लिकेशन प्रोग्रामिंग भाषा के मौजूदा प्राइमेटिव्स से शुरू करके और धीरे-धीरे अधिक से अधिक जटिल सुविधाओं को जोड़कर बनाया जाता है जब तक कि संपूर्ण एप्लिकेशन लिखा नहीं जाता है। टॉप-डाउन प्रोग्रामिंग पर बॉटम-अप प्रोग्रामिंग के कई फायदे हैं। निम्नलिखित नुसार:

परीक्षण को सरल बनाया गया है क्योंकि स्टब्स की आवश्यकता नहीं है।

टेस्ट फंक्शन स्टब्स की तुलना में आसान होते हैं और एक इंटरैक्टिव प्रोग्रामिंग वातावरण, यानी कॉमन लिस्प या जीडीबी के मामले में आवश्यक होते हैं।

बॉटम-अप प्रोग्रामिंग अधिक सामान्य होती है, और इस प्रकार टॉप-डाउन प्रोग्रामिंग की तुलना में अधिक पुनः प्रयोज्य होती है।

इसका उपयोग एप्लिकेशन-विशिष्ट भाषा बनाने के लिए किया जा सकता है।

इस प्रकार की भाषा अनुप्रयोगों के पूरे वर्ग को लागू करने के लिए उपयुक्त है, न कि केवल लेखन के लिए आवश्यक।

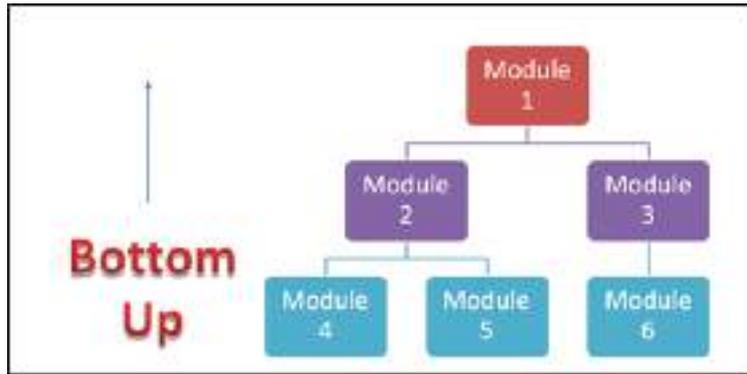
यह रखरखाव को बहुत सरल करता है, खासकर जब एप्लिकेशन में नई सुविधाएं जोड़ते हैं।

यह डेवलपर्स को एप्लिकेशन की सटीक कार्यक्षमता पर अंतिम निर्णय को स्थगित करने की भी अनुमति देता है।

इस निर्णय को स्थगित करने की क्षमता इस संभावना को कम कर देती है कि ग्राहक आवेदन के विनिर्देशों की स्थापना और इसके कार्यान्वयन के बीच अपना मन बदल लेगा।

### बॉटम-अप प्रोग्रामिंग पर काम करना

सी या जावा (C or Java) जैसी भाषा में बॉटम-अप प्रोग्रामिंग में भाषा आदिम या मौजूदा सार डेटा प्रकारों से सार डेटा प्रकार बनाना शामिल है।



चित्र 3.4 बॉटम उप डिजाईन

(Image Source: <https://www.guru99.com/integration-testing.html>)

टॉप-डाउन डिज़ाइन और बॉटम-अप डिज़ाइन के बीच अंतर

क्रमांक	कोई टॉप-डाउन अप्रोच	बॉटम-अप अप्रोच
1.	इस दृष्टिकोण में, हम समस्या को छोटे भागों में तोड़ने पर ध्यान केंद्रित करते हैं।	बॉटम-अप दृष्टिकोण में, हम छोटी समस्याओं को हल करते हैं और उन्हें समय रूप से एकीकृत करते हैं और समाधान को पूरा करते हैं।
2.	मुख्य रूप से संरचित प्रोग्रामिंग भाषाओं जैसे CO-BOL, Fortran, C, आदि द्वारा उपयोग किया जाता है।	मुख्य रूप से ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषाओं जैसे C++, C#, Python . द्वारा उपयोग किया जाता है
3.	प्रत्येक भाग को अलग से क्रमादेशित किया जाता है और इसलिए इसमें अतिरेक होता है।	डेटा एनकैप्सुलेशन और डेटा छिपाने का उपयोग करके अतिरेक को कम किया जाता है।
4.	इसमें मॉड्यूल के बीच कम्युनिकेशन कम होता है।	इस मॉड्यूल में संचार होना चाहिए।
5.	इसका उपयोग डिबगिंग, मॉड्यूल प्रलेखन आदि में किया जाता है।	इसका उपयोग परीक्षण में किया जाता है।
6.	ऊपर से नीचे के दृष्टिकोण में, अपघटन होता है।	नीचे-ऊपर के दृष्टिकोण में, रचना होती है।
7.	सिस्टम के इस शीर्ष कार्य को पहचानना कठिन हो सकता है।	कभी-कभी हम अपने द्वारा शुरू किए गए टुकड़े से एक कार्यक्रम नहीं बना सकते हैं।

चित्र 3.2 टॉप-डाउन डिज़ाइन और बॉटम-अप डिज़ाइन के बीच अंतर

### 3.1.4 डिजाइन परीक्षण रणनीतियाँ

परीक्षण डिजाइन तकनीकों के प्रकार निम्न हैं:

**स्वैच्छिक परीक्षण**

- धारमिक चरण परीक्षण तकनीक स्वैच्छिक परीक्षण है।
- यह पासक में निश्चित उत्पाद को चलाए बिना किया जाता है।
- अनिर्धार्य रूप से, कोड में ही दोषों और त्रुटियों को खोजने के लिए इस प्रकार की टेस्ट-जांच आवश्यक है।
- इस तरह की एक पूर्व-तैयारी जवाब महत्वपूर्ण है क्योंकि यह कोड दोषों और सॉफ्टवेयर सराफा दोषों के कारण होने वाली कठिनाइयों को दूर करने में मदद करती है।

**संरचनात्मक परीक्षण**

- सॉफ्टवेयर को पहले चलाए बिना उसका संरचनात्मक परीक्षण करना असंभव है।
- संरचनात्मक परीक्षण, जिसे एंटी-बॉक्स परीक्षण के रूप में भी जाना जाता है, सॉफ्टवेयर विकास प्रक्रिया के पूर्व-उत्पादन चरण के दौरान उत्पादन होने वाली त्रुटियों और कमियों को पता लगाने और उन्हें ठीक करने के लिए आवश्यक है।
- इस बिंदु पर, कार्यक्रम संरचना के आधार पर बड़े परीक्षण करने के लिए प्रतिगमन परीक्षण का उपयोग किया जाता है।
- अधिकतर समय, यह एक स्वचालित प्रक्रिया है जो इस बिंदु पर विकास प्रक्रिया को रूढ़ि देने में मदद करने के लिए एक परीक्षण स्वचालन टाचे के भीतर चलाए जाती है।
- इंटरफ़ेस और न्यूए इंजीनियरी के पास सॉफ्टवेयर की संरचना और डेटा प्रवाह (डेटा प्रवाह परीक्षण) तक पूरी पहुंच है, जिससे उन्हें परीक्षण के परिणामों की तुलना पहले के फुलरप्रतिष्ठों के परिणामों (नियंत्रण प्रवाह परीक्षण) से सिस्टम के व्यवहार (न्यूएशन परीक्षण) में किसी भी बदलाव का पता लगाने की अनुमति मिलती है। )

**व्यवहार परीक्षण**

- परीक्षण का अंतिम चरण इन प्रतिक्रियाओं को अंतर्निहित तंत्र के बजाय विभिन्न गतिविधियों के लिए सॉफ्टवेयर की प्रतिक्रियाओं पर केंद्रित है।
- दूसरे शब्दों में, व्यवहार परीक्षण, जिसे ब्लैक-बॉक्स परीक्षण के रूप में भी जाना जाता है, उपयोगकर्ता के दृष्टिकोण से उत्पाद की जांच करने के लिए बड़े परीक्षण करता है, जिसमें से अधिकांश मैनुअल हैं।
- उपयोगिता परीक्षण सफलता के लिए, उत्पादन के लिए, और उत्पाद के निश्चित उपयोगकर्ताओं के रूप में धरा पर प्रतिक्रिया करने के लिए, न्यूए इंजीनियरी के पास आमतौर पर किसी व्यवसाय या सॉफ्टवेयर के अन्य उद्देश्यों (ब्लैक बॉक्स) के बारे में कुछ विशिष्ट जानकारी होती है।
- यदि दोहराव वाली गतिविधियों की आवश्यकता होती है, तो मूलतः त्रुटि को समाप्त करने के लिए व्यवहार परीक्षण में स्वचालन (प्रतिगमन परीक्षण) भी शामिल हो सकते हैं।
- उत्पादन के लिए, यह देखने के लिए कि उत्पाद इस तरह की गतिविधि को कैसे संभालता है, आपको वेबसाइट पर 100 पंजीकरण फॉर्म भरने पड़ सकते हैं।

चित्र 3.5 टेस्ट डिजाइन तकनीकों के प्रकार

सबसे अच्छी परीक्षण रणनीति आमतौर पर वह होती है जो विशिष्ट परियोजना के लिए पूरी तरह से अच्छी तरह से काम करती है। हालांकि, एक प्रभावी रणनीति विकसित करने के प्रमुख सिद्धांत हैं:



चित्र 3.6 एक प्रभावी रणनीति विकसित करने के प्रमुख सिद्धांत

### 3.1.5 अर्थ संबंधी त्रुटियां

अर्थ संबंधी त्रुटियां सूक्ष्म प्रकार की गलतियां हैं। सिमेंटिक त्रुटि तब होती है जब कोई कथन वाक्य-रचना की दृष्टि से सही होता है लेकिन प्रोग्रामर द्वारा इच्छित कार्य को निष्पादित नहीं करता है। चूंकि निर्माण भाषा के नियमों का पालन करता है, इसलिए संकलक या दुभाषिया द्वारा अर्थ संबंधी त्रुटियों का पता नहीं लगाया जाता है। एक प्रोग्रामिंग त्रुटि एक प्रोग्रामिंग भाषा निर्माण के अर्थ या प्रभाव की गलत व्याख्या के कारण होती है।

सिमेंटिक एरर के कारण: जब सिमेंटिक या लेक्सिकल रिप्रेजेंटेशन तक पहुंच बाधित होती है, तो सिमेंटिक एरर होता है। इन त्रुटियों की उत्पत्ति का निर्धारण करने का एक तरीका उन शब्दों की समझ का मूल्यांकन करना है जो नामकरण में अर्थ संबंधी त्रुटियों को उजागर करते हैं।

## सारांश

- सॉफ्टवेयर डेवलपमेंट प्रक्रिया चरणों की एक श्रृंखला है जिसे डिलिवरेबल्स के उत्पादन के लिए पूरा किया जाना चाहिए।
- सॉफ्टवेयर डेवलपमेंट प्रक्रिया, एसडीएलसी के विपरीत, प्रत्येक चरण को निर्दिष्ट करती है जिसे डिलिवरेबल्स के हिस्से के रूप में पूरा किया जाना चाहिए जिसमें प्रत्येक चरण शामिल है। यह छोटे पैमाने पर चरणों की एक श्रृंखला है।
- व्यापार विश्लेषक और परियोजना प्रबंधक ने ग्राहक के साथ एक बैठक की स्थापना की ताकि सभी जानकारी एकत्र की जा सके जैसे ग्राहक क्या बनाना चाहता है, अंतिम उपयोगकर्ता कौन होगा, और उत्पाद का उद्देश्य क्या है। उत्पाद बनाने से पहले उत्पाद की मूल समझ या ज्ञान बहुत महत्वपूर्ण है।
- टॉप-डाउन प्रोग्रामिंग बॉटम-अप प्रोग्रामिंग के विपरीत है। यह प्रोग्रामिंग की एक शैली को संदर्भित करता है जहां एक एप्लिकेशन का निर्माण किया जाता है, जो उच्च-स्तरीय विवरण के साथ शुरू होता है कि इसे क्या करना चाहिए और विनिर्देश को सरल और सरल टुकड़ों में तोड़ना जब तक कि एक स्तर तक नहीं पहुंच जाता है जो कि प्राइमेटिव से मेल खाती है प्रोग्रामिंग की इस्तेमाल की जाने वाली भाषा।
- बॉटम-अप प्रोग्रामिंग टॉप-डाउन प्रोग्रामिंग के विपरीत है। यह प्रोग्रामिंग की एक शैली को संदर्भित करता है जहां एक एप्लिकेशन का निर्माण प्रोग्रामिंग भाषा के मौजूदा प्राइमेटिव से शुरू होता है और धीरे-धीरे अधिक से अधिक जटिल सुविधाओं का निर्माण करता है जब तक कि सभी एप्लिकेशन लिखे नहीं जाते हैं।
- सी या जावा जैसी भाषा में, बॉटम-अप प्रोग्रामिंग भाषा के आदिम या मौजूदा सार डेटा प्रकारों से अमूर्त डेटा प्रकारों के निर्माण का रूप लेती है।
- टॉप-डाउन प्रोग्रामिंग आमतौर पर कार्यों या प्रक्रियाओं के रूप में, कार्यक्षमता पर आधारित मॉड्यूल उत्पन्न करने के लिए जाता है। आमतौर पर, सिस्टम का उच्च-स्तरीय विनिर्देश कार्यक्षमता बताता है।
- बॉटम-अप दृष्टिकोण में, हम छोटी समस्याओं को हल करते हैं और उन्हें समग्र रूप से एकीकृत करते हैं और समाधान को पूरा करते हैं।
- ऊपर से नीचे के दृष्टिकोण में, अपघटन होता है।
- बॉटम-अप अप्रोच में कंपोजिशन होता है।
- सी या जावा जैसी भाषा में, बॉटम-अप प्रोग्रामिंग भाषा के आदिम या मौजूदा सार डेटा प्रकारों से अमूर्त डेटा प्रकारों के निर्माण का रूप लेती है।

## अभ्यास



निम्नलिखित प्रश्नों के उत्तर दो:

सॉफ्टवेयर डेवलपमेंट प्रक्रिया की व्याख्या कीजिए।

सिमेंटिक एरर क्या हैं?

एक प्रभावी रणनीति विकसित करने के प्रमुख सिद्धांत क्या हैं?

डिजाइन परीक्षण रणनीतियों का वर्णन करें।

टॉप-डाउन डिजाइन और बॉटम-अप डिजाइन के क्या लाभ हैं?

रिक्त स्थान भरें

\_\_\_\_\_ बॉटम-अप प्रोग्रामिंग के विपरीत है।

\_\_\_\_\_ जैसी भाषा में, बॉटम-अप प्रोग्रामिंग भाषा के आदिम या मौजूदा अमूर्त डेटा प्रकारों से अमूर्त डेटा प्रकारों के निर्माण का रूप लेती है।

ग) \_\_\_\_\_ डेवलपमेंट के पूरे चक्र को परिभाषित करता है अर्थात उत्पाद के रखरखाव के लिए एक आवश्यकता को इकट्ठा करने में शामिल सभी कार्य।

\_\_\_\_\_ एक ढांचा है जो सॉफ्टवेयर के डेवलपमेंट में शामिल चरणों को परिभाषित करता है। इसमें सॉफ्टवेयर के निर्माण, परिनियोजन और रखरखाव की विस्तृत योजना शामिल है।



## 4. अनुप्रयोग (एप्लीकेशन) प्रबंधन की प्रक्रिया



IT - ITeS SSC  
NASSCOM

- यूनिट 4.1 सॉफ्टवेयर आवश्यकताएँ विशिष्टता की गुणवत्ता विशेषताएँ  
यूनिट 4.2 कस्टम और रैपिड एप्लीकेशन सॉफ्टवेयर



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. कार्यो को स्वचालित करने और सरल प्रोग्राम लिखने के लिए स्क्रिप्टिंग भाषा के उचित अनुप्रयोग का उपयोग करने में
2. एक अच्छे एसआरएस की सॉफ्टवेयर गुणवत्ता विशेषताओं और विशेषताओं की सूची बनाने में

## यूनिट 4.1: सॉफ्टवेयर आवश्यकताएँ विशिष्टता की गुणवत्ता विशेषताएँ

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. आवश्यकता विश्लेषण के लिए उपयोग की जाने वाली विभिन्न तकनीकों की पहचान करने में

### 4.1.1 सॉफ्टवेयर आवश्यकताएँ विशिष्टता और विश्लेषण

एक सॉफ्टवेयर आवश्यकता विनिर्देश (एसआरएस) एक दस्तावेज है जो निर्दिष्ट करता है कि सॉफ्टवेयर क्या करेगा और इसे कैसे कार्य करना चाहिए। यह उस कार्यक्षमता का भी वर्णन करता है जो उत्पाद को सभी हितधारकों (व्यवसाय और उपयोगकर्ताओं) की जरूरतों को पूरा करना चाहिए। SRS को ग्राहक और ठेकेदारों के बीच हुए समझौते के अनुसार विकसित किया गया है। इसमें उपयोग के मामले हो सकते हैं जो यह बताते हैं कि उपयोगकर्ता सॉफ्टवेयर सिस्टम के साथ कैसे इंटरैक्ट करेगा।

एसआरएस आवश्यकताएँ: सिस्टम इंजीनियरों को एक अधिग्रहण कार्यक्रम के पूरे जीवन में कई तरह की आवश्यकताओं का निर्माण करना चाहिए। ये आवश्यकताएँ उच्च-स्तरीय अवधारणा-केंद्रित से लेकर भाग-विशिष्ट तक होती हैं।



चित्र 4.1 SRS के लिए विशेषताएँ

1. कार्यात्मक आवश्यकताएँ: एक कार्यात्मक आवश्यकता केवल एक कार्य है (जिसे एक क्रिया या गतिविधि के रूप में भी जाना जाता है) जिसे एक परिचालन क्षमता प्रदान करने के लिए पूरा किया जाना चाहिए (या एक परिचालन आवश्यकता को पूरा करना)। कुछ परिचालन और समर्थन कार्यात्मक आवश्यकताओं को आवश्यक परिचालन क्षमता से घटाया जा सकता है। सिस्टम इंजीनियरिंग अनुभव के अनुसार, डेवलपमेंट, निर्माण, सत्यापन, परिनियोजन, प्रशिक्षण, संचालन, समर्थन और निपटान आठ सामान्य कार्य हैं जिन्हें अधिकांश प्रणालियों को अपने जीवन चक्र में पूरा करना चाहिए। इन्हें "आठ प्राथमिक सिस्टम फ़ंक्शंस" के रूप में संदर्भित किया जाता है। सिस्टम के लिए सभी कार्यात्मक आवश्यकताओं की पहचान करते समय प्रत्येक को आमतौर पर ध्यान में रखा जाना चाहिए।

2. प्रदर्शन आवश्यकताएँ: एक प्रदर्शन आवश्यकता एक बयान है जो निर्दिष्ट करता है कि किसी कार्य को किस हद तक किया जाना चाहिए, आमतौर पर मात्रा, सटीकता, कवरेज, समयबद्धता, या तत्परता के संदर्भ में। प्रचालनात्मक कार्य के लिए प्रदर्शन आवश्यकताएँ और, कभी-कभी, कुछ अन्य अक्सर संयुक्त क्षमता एकीकरण और डेवलपमेंट प्रणाली (जेसीआईडीएस) प्रक्रिया द्वारा विकसित आवश्यक परिचालन क्षमता के बयान के साथ अच्छी तरह से संबंध रखते हैं। अन्य प्रदर्शन आवश्यकताओं के लिए आमतौर पर व्यापक सिस्टम इंजीनियरिंग की आवश्यकता होती है।

3. सिस्टम तकनीकी आवश्यकताएँ: परिणामस्वरूप, आवंटित और व्युत्पन्न दोनों आवश्यकताएँ उत्पन्न होती हैं।

a. आवंटित आवश्यकताएँ: सिस्टम आवश्यकताओं से सीधे सिस्टम तत्वों तक प्रवाहित होती हैं।

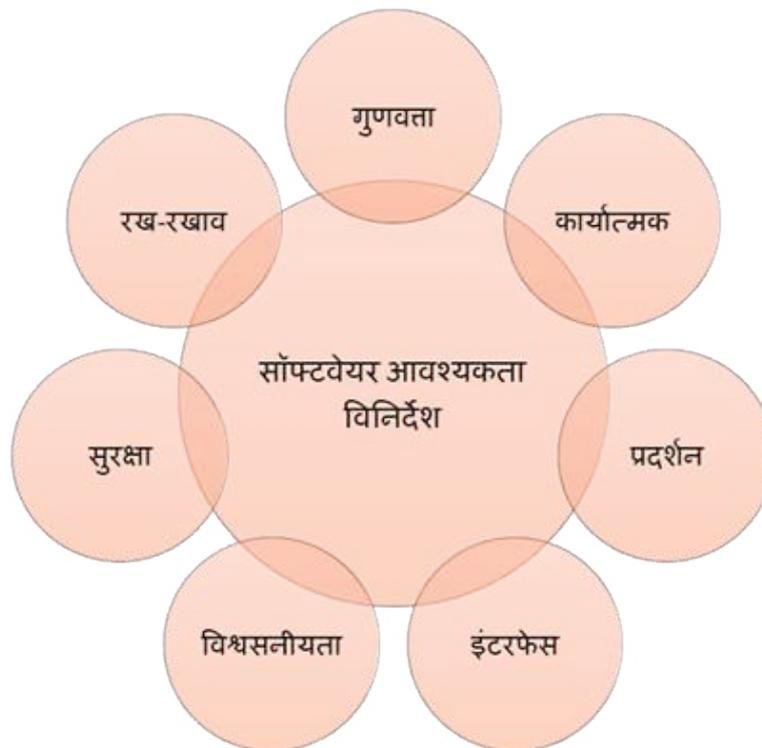
b. डिज़ाइन समाधान से प्राप्त आवश्यकताएँ (और इसलिए कभी-कभी डिज़ाइन आवश्यकताएँ कहलाती हैं)। उनमें सिस्टम तत्वों के बीच आंतरिक इंटरफ़ेस बाधाएँ शामिल हैं।

4. निर्दिष्टीकरण: एक विनिर्देश विवरणों का एक विस्तृत, सटीक विवरण है, विशेष रूप से वह जो किसी चीज के निर्माण, स्थापित या निर्मित होने के लिए सामग्री, आयाम और कारीगरी की गुणवत्ता को निर्दिष्ट करता है। एक विनिर्देश का समग्र लक्ष्य एक उत्पाद या सेवा प्राप्त करने के लिए एक आधार प्रदान करना है जो उचित लागत पर एक विशिष्ट आवश्यकता को पूरा करेगा और अधिकतम उचित प्रतिस्पर्धा को प्रोत्साहित करेगा। एक विनिर्देश, परिभाषा के अनुसार, सीमा निर्धारित करता है और, परिणामस्वरूप, उन वस्तुओं को समाप्त या संभावित रूप से समाप्त करता है जो खींची गई सीमाओं से बाहर हैं। एक अच्छे विनिर्देशन को निम्नलिखित चार लक्ष्यों को पूरा करना चाहिए:



चित्र 4.2 अच्छी विशिष्टता के लिए लक्ष्य

#### 4.1.2 सॉफ्टवेयर आवश्यकताओं की विशिष्टता की विशेषताएँ



चित्र 4.3 सॉफ्टवेयर आवश्यकता विनिर्देश

एक अच्छे SRS की गुणवत्ता विशेषताएँ

एक अच्छे SRS दस्तावेज़ की विशेषताएँ निम्नलिखित हैं:

1. पूर्ण - एक पूर्ण और सटीक आवश्यकता विनिर्देश को उन सभी वास्तविक-विश्व परिदृश्यों को सटीक रूप से परिभाषित करना चाहिए जिनका सामना किया जाएगा, साथ ही साथ क्षमता की प्रतिक्रियाएँ भी। इसमें ऐसी स्थितियाँ या क्षमताएँ शामिल नहीं होनी चाहिए जिनका उपयोग नहीं किया जाएगा।
2. संगत - सिस्टम फ़ंक्शंस और प्रदर्शन स्तर संगत होना चाहिए, और आवश्यक गुणवत्ता सुविधाएँ (विश्वसनीयता, सुरक्षा, सुरक्षा, और इसी तरह) सिस्टम की उपयोगिता के साथ असंगत नहीं होनी चाहिए। उदाहरण के लिए, एकमात्र सुरक्षित विमान वह है जिसे एनोटेट किया गया है, जिसमें कोई ईंधन या अन्य तरल पदार्थ नहीं है, और सुरक्षित रूप से बंधे हैं।
3. सही - विनिर्देश को वांछित क्षमता के वास्तविक दुनिया के परिचालन वातावरण, उस वातावरण के लिए उसके इंटरफ़ेस और उस वातावरण के साथ उसकी बातचीत को परिभाषित करना चाहिए। विनिर्देश शुद्धता प्राप्त करने का सबसे कठिन पहलू आवश्यकताओं का वास्तविक-विश्व पहलू है। नए अनुप्रयोगों के लिए, वास्तविक दुनिया का वातावरण अज्ञात है, और परिपक्व अनुप्रयोगों के लिए, वास्तविक दुनिया का वातावरण लगातार बदल रहा है। Y2K समस्या, जो 1999 से 2000 के संक्रमण के दौरान हुई, वास्तविक दुनिया का एक उदाहरण है जो किसी एप्लिकेशन की निर्दिष्ट आवश्यकताओं से अधिक है।
4. परिवर्तनीय - संबंधित चिंताओं को समूहित करें, जबकि असंबंधित को अलग किया जाना चाहिए। परिवर्तनीय होने के लिए, आवश्यकता दस्तावेज़ों में एक तार्किक संरचना होनी चाहिए।
5. रैंक - आवश्यकताओं के दस्तावेज़ का संगठन और संरचना स्थिरता और महत्व के आधार पर विनिर्देश विवरण की रैंकिंग स्थापित करती है। आवश्यकता विनिर्देश द्वारा संबोधित समस्या जितनी बड़ी और अधिक जटिल होती है, उतना ही कठिन एक दस्तावेज़ बनाना होता है जो समझने में बाधा डालने के बजाय सहायता करता है।
6. परीक्षण योग्य - एक आवश्यकता विनिर्देश को इस तरह से लिखा जाना चाहिए जो इसे विनिर्देश और संदर्भ जानकारी से प्राप्त पास, असफल, या मात्रात्मक मूल्यांकन मानदंड के खिलाफ परीक्षण करने की अनुमति देता है। आवश्यकता है कि एक प्रणाली का उपयोग करने के लिए "आसान" है व्यक्तिपरक है और इस प्रकार अनुपयुक्त है।
7. ट्रेस करने योग्य - एसआरएस दस्तावेज़ में बताई गई प्रत्येक आवश्यकता को ट्रेसबिलिटी सुनिश्चित करने के लिए विशिष्ट रूप से पहचाना जाना चाहिए। आवश्यकताओं के दस्तावेज़ के भीतर प्रत्येक विनिर्देश विवरण को पहचान प्रदान करने के लिए एक सुसंगत और तार्किक योजना का उपयोग विशिष्टता में सहायता करता है।
8. असंदिग्ध - एक आवश्यकता कथन असंदिग्ध है यदि इसकी केवल एक तरह से व्याख्या की जा सकती है। प्राकृतिक भाषा के माध्यम से प्राप्त करना संभवतः सबसे कठिन विशेषता है। कमजोर वाक्यांशों या खराब वाक्य संरचना के उपयोग से विनिर्देश विवरण में गलतफहमियाँ पैदा होंगी।
9. मान्य - एक आवश्यकता विनिर्देश को मान्य करने के लिए, प्रबंधकों, इंजीनियरों और ग्राहक प्रतिनिधियों सहित सभी परियोजना प्रतिभागियों को इसे समझने, विश्लेषण करने और स्वीकार या अस्वीकार करने में सक्षम होना चाहिए। यह प्राथमिक कारण है कि अधिकांश विनिर्देश प्राकृतिक भाषा में लिखे गए हैं।
10. सत्यापन योग्य - अमूर्तता के एक स्तर पर आवश्यकता विनिर्देश सत्यापन योग्य होने के लिए अमूर्तता के दूसरे स्तर पर संगत होना चाहिए। इन विशेषताओं में से अधिकांश, यदि सभी नहीं हैं, तो व्यक्तिपरक हैं, और आवश्यकताओं के विनिर्देश की गुणवत्ता का निर्धारण करने के लिए आवश्यकताओं द्वारा संबोधित डोमेन में तकनीकी और परिचालन विशेषज्ञों द्वारा समीक्षा और विश्लेषण की आवश्यकता होती है।



- यथार्थता
- संपूर्णता
- संगतता
- टेस्टेबिलिटी
- डिजाइन स्वतंत्रता
- अमूर्तता का सही स्तर
- परिवर्तनीयता
- पता लगाने की क्षमता
- महत्व के लिए रैंकिंग
- ग्राहक द्वारा समझा जा सकता है
- सत्यापनीयता
- असदिग्धता

चित्र 4.4 एक अच्छे SRS के लक्षण

एसआरएस दस्तावेजों के उपयोग

SRS दस्तावेजों के कुछ प्रमुख उपयोग निम्नलिखित हैं:



चित्र 4.5 एसआरएस दस्तावेजों के उपयोग

### 4.1.3 एसआरएस के लिए IEEE मानक

आवश्यकता दस्तावेज़ को लिखने, समीक्षा करने और बनाए रखने में आसान होने के लिए डिज़ाइन किया गया है। इसे खंडों में और आगे मॉड्यूल या इकाइयों में विभाजित किया गया है। यह ध्यान दिया जाना चाहिए कि एसआरएस में शामिल विवरण का स्तर सिस्टम प्रकार और डेवलपमेंट के लिए उपयोग किए जाने वाले प्रक्रिया मॉडल द्वारा निर्धारित किया जाता है। उदाहरण के लिए, यदि किसी बाहरी ठेकेदार को सिस्टम विकसित करना है, तो महत्वपूर्ण सिस्टम विनिर्देश सटीक और विस्तृत होने चाहिए। इसी तरह, जब आवश्यकताओं को लचीला होना चाहिए और आंतरिक विकास का उपयोग किया जाता है, तो आवश्यकता दस्तावेज़ कम विस्तृत हो सकते हैं।

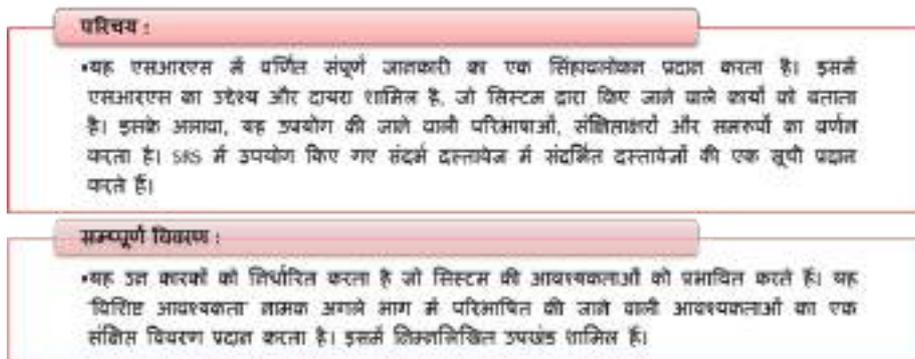
चूंकि आवश्यकता दस्तावेज़ बाद के सॉफ्टवेयर डेवलपमेंट चरणों की नींव के रूप में कार्य करता है, इसलिए यह महत्वपूर्ण है कि इसे निर्धारित तरीके से विकसित किया जाए। इस उद्देश्य के लिए एसआरएस तैयार करते समय कुछ दिशानिर्देशों का पालन किया जाता है। इन नियमों की रूपरेखा नीचे दी गई है।



चित्र 4.6 एसआरएस दस्तावेज़ों के लिए नियम और दिशानिर्देश

SRS में शामिल सूचना का प्रकार कई कारकों द्वारा निर्धारित किया जाता है, जिसमें विकसित किए जा रहे सॉफ्टवेयर के प्रकार और इसके विकास में उपयोग किए जाने वाले दृष्टिकोण शामिल हैं। पुनरावृत्त विकास प्रक्रिया का उपयोग करके विकसित सॉफ्टवेयर के लिए आवश्यकता दस्तावेज़ महत्वपूर्ण प्रणालियों के लिए विकसित सॉफ्टवेयर की तुलना में कम विस्तृत होगा। यह इस तथ्य के कारण है कि इन प्रणालियों में विनिर्देश अत्यंत विस्तृत और सटीक होने चाहिए। एक आवश्यकता दस्तावेज़ के विकास के लिए कई मानक प्रस्तावित किए गए हैं। हालांकि, सबसे व्यापक रूप से इस्तेमाल किया जाने वाला मानक आईईईईई द्वारा है, जो एक सामान्य ढांचे के रूप में कार्य करता है। इस सामान्य ढांचे को एक विशिष्ट संगठन की जरूरतों को पूरा करने के लिए तैयार किया जा सकता है।

प्रत्येक एसआरएस एक विशिष्ट पैटर्न का अनुसरण करता है; इस प्रकार, इसे समझना आसान बनाने के लिए आवश्यकता दस्तावेज़ की संरचना को मानकीकृत करना महत्वपूर्ण है। यह आईईईईई मानक एसआरएस के लिए विभिन्न परियोजनाओं के लिए आवश्यकताओं को व्यवस्थित करने के लिए उपयोग किया जाता है, एसआरएस की संरचना के विभिन्न तरीके प्रदान करता है। सभी आवश्यकताओं के दस्तावेज़ों के पहले दो खंड समान हैं। इस दस्तावेज़ के खंड इस प्रकार हैं।



#### उत्पाद परिप्रेक्ष्य:

- यह निर्धारित करता है कि उत्पाद एक स्वतंत्र उत्पाद है या बड़े उत्पाद का अभिन्न अंग है। यह हार्डवेयर, सॉफ्टवेयर, सिस्टम और संचार के साथ इंटरफेस को निर्धारित करता है। यह उपयोगकर्ता द्वारा उपयोग की जाने वाली स्मृति बाधाओं और संचालन को भी परिभाषित करता है।

#### प्रोडक्ट कार्य :

- यह सॉफ्टवेयर द्वारा किए जाने वाले कार्यों का सारांश प्रदान करता है। कार्यों को एक सूची में व्यवस्थित किया जाता है ताकि उपयोगकर्ता द्वारा उन्हें आसानी से समझा जा सके:

#### उपभोगता विशेषताएं :

- यह उपयोगकर्ताओं की सामान्य विशेषताओं को निर्धारित करता है।

#### रूकावट :

- यह नियामक नीतियों, ऑडिट कार्यों, विश्वसनीयता आवश्यकताओं, आदि जैसी बाधाओं का जेनेरा 1 विवरण प्रदान करता है।

#### धारणा और निर्भरता:

- यह मान्यताओं और कारकों की एक सूची प्रदान करता है जो इस दस्तावेज में बताई गई आवश्यकताओं को प्रभावित करते हैं।

#### आवश्यकताओं का बंटवारा:

- यह उन आवश्यकताओं को निर्धारित करता है जिन्हें सिस्टम के भविष्य के संस्करणों के जारी होने तक विलंबित किया जा सकता है।

#### विशिष्ट जरूरतें:

- ये सभी आवश्यकताओं को विस्तार से निर्धारित करते हैं ताकि डिजाइनर उनके द्वारा सिस्टम को डिजाइन कर सकें। आवश्यकताओं में सिस्टम के प्रत्येक इनपुट और आउटपुट और प्रदान किए गए इनपुट के जवाब में किए गए कार्यों का विवरण शामिल है। इसमें निम्नलिखित उपखंड शामिल हैं।

#### बाह्य इंटरफेस:

- यह अन्य सिस्टम के साथ सॉफ्टवेयर के इंटरफेस को निर्धारित करता है, जिसमें सिस्टम के साथ इंटरफेस आदि शामिल हो सकते हैं। बाहरी इंटरफेस उपयोगकर्ताओं, हार्डवेयर या अन्य सॉफ्टवेयर के साथ सॉफ्टवेयर की सहभागिता को भी निर्दिष्ट करता है। सॉफ्टवेयर उत्पाद के प्रत्येक उपयोगकर्ता इंटरफेस की विशेषताएं SRS में निर्दिष्ट हैं। हार्डवेयर इंटरफेस के लिए, SRS सॉफ्टवेयर और हार्डवेयर घटकों के बीच प्रत्येक इंटरफेस की तार्किक विशेषताओं को निर्दिष्ट करता है। यदि सॉफ्टवेयर को मौजूदा हार्डवेयर पर निष्पादित किया जाता है, तो स्मृति प्रतिबंध जैसी विशेषताओं को भी निर्दिष्ट किया जाता है।

#### कार्य :

- यह प्रणाली की कार्यात्मक क्षमताओं को निर्धारित करता है। प्रत्येक कार्यात्मक आवश्यकता के लिए, आउटपुट उत्पन्न करने के लिए इनपुट की स्वीकृति और प्रसंस्करण निर्दिष्ट है। इसमें इनपुट पर वैधता जांच, संचालन का सटीक क्रम, आउटपुट के लिए इनपुट का संबंध आदि

#### प्रदर्शन संबंधी जरूरतें:

- **Performance constraints of the software system.** Performance आवश्यकता दो प्रकार की होती है: स्थिर आवश्यकताएं और गतिशील आवश्यकताएं। स्थिर आवश्यकताएं (क्षमता आवश्यकताओं के रूप में भी जानी जाती हैं) सिस्टम की निष्पादन विशेषताओं पर प्रतिबंध नहीं लगाती हैं। इनमें टर्मिनलों की संख्या और समर्थित उपयोगकर्ताओं जैसी आवश्यकताएं शामिल हैं। गतिशील आवश्यकताएं सिस्टम के व्यवहार के निष्पादन पर बाधाओं को निर्धारित करती हैं, जिसमें प्रतिक्रिया समय (निर्दिष्ट शर्तों के तहत एक ऑपरेशन की शुरुआत और समाप्ति के बीच का समय) और थ्रूपुट (कुल) शामिल है (amount of work done in a given time).

**आवश्यकताओं का तार्किक डेटाबेस:**

- यह डेटाबेस में संग्रहित करने के लिए तार्किक आवश्यकताओं को निर्धारित करता है। इसमें उपयोग की गई जानकारी का प्रकार, उपयोग की आवृत्ति, डेटा निकाय और उनके बीच संबंध आदि शामिल हैं।

**डिजाइन बाधा:**

- यह सभी डिजाइन बाधाओं को निर्धारित करता है जो मानकों, हार्डवेयर सीमाओं आदि द्वारा लगाए जाते हैं। मानक अनुपालन प्रणाली के लिए आवश्यकताओं को निर्धारित करता है, जो निर्दिष्ट मानकों का अनुपालन करता है। इन मानकों में लेखांकन प्रक्रियाएं और रिपोर्ट प्रारूप शामिल हो सकते हैं। हार्डवेयर सीमाओं का अर्थ तब होता है जब सॉफ्टवेयर मौजूदा हार्डवेयर या कुछ पूर्व-निर्धारित हार्डवेयर पर काम कर सकता है। यह सॉफ्टवेयर डिजाइन विकसित करते समय प्रतिबंध लगा सकता है। हार्डवेयर सीमाओं में मशीन का हार्डवेयर कॉन्फिगरेशन और उपयोग किए जाने वाले ऑपरेटिंग सिस्टम शामिल हैं।

**सॉफ्टवेयर सिस्टम विशेषताएँ:**

- यह विश्वसनीयता, उपलब्धता, रखरखाव और पोर्टेबिलिटी जैसी विशेषताएं प्रदान करता है। यह सत्यापित करने के लिए कि वे अंतिम प्रणाली में प्राप्त किए गए हैं, इन सभी विशेषताओं का वर्णन करना आवश्यक है।

**विशिष्ट आवश्यकताओं का आयोजन:**

- यह आवश्यकताओं को निर्धारित करता है ताकि उन्हें इष्टतम समझ के लिए ठीक से व्यवस्थित किया जा सके। आवश्यकताओं को संचालन के तरीके, उपयोगकर्ता वर्गों, वस्तुओं, सुविधाओं, प्रतिक्रिया और कार्यात्मक पदानुक्रम के आधार पर व्यवस्थित किया जा सकता है।

**प्रबंधन प्रक्रिया बदलें:**

- यह परियोजना के दायरे और आवश्यकताओं में परिवर्तन को प्रतिबिंबित करने के लिए एसआरएस की पहचान, मूल्यांकन और अद्यतन करने के लिए परिवर्तन प्रबंधन प्रक्रिया को निर्धारित करता है।

**दस्तावेज़ अनुमोदन:**

- ये एसआरएस दस्तावेज़ के अनुमोदनकर्ताओं के बारे में जानकारी प्रदान करते हैं जैसे कि अनुमोदनकर्ता का नाम, हस्ताक्षर, तिथि

**सहायक सूचना:**

- यह सामग्री तालिका, अनुक्रमणिका आदि जैसी जानकारी प्रदान करता है। यह विशेष रूप से तब आवश्यक है जब एसआरएस बड़ी और जटिल परियोजनाओं के लिए तैयार किया जाता है।

**सॉफ्टवेयर गुणवत्ता गुण**

- उपलब्धता: उड़ान निर्दिष्ट तिथि और निर्दिष्ट समय पर उपलब्ध होनी चाहिए क्योंकि कई ग्राहक आरक्षण कर रहे हैं।
- शुद्धता: उड़ान सही स्टार्ट टर्मिनल से शुरू होकर पहुंचना चाहिए और सही गंतव्य तक पहुंचना चाहिए।
- रख-रखाव: व्यवस्थापकों और फ्लाइट चार्जर्स को फ्लाइट का सही शेड्यूल बनाए रखना चाहिए।
- प्रयोज्यता: उड़ान शेड्यूल को ग्राहकों की ज़्यादा से ज़्यादा ज़रूरतों को पूरा करना चाहिए।

चित्र 4.7 एसआरएस दस्तावेजों और उसके घटकों के लिए मानक संरचना

### 4.1.4 संरचित डिजाइन

संरचित विश्लेषण के दौरान, सिस्टम के प्रमुख प्रसंस्करण कार्यों (अर्थात कार्यों) का विश्लेषण किया जाता है, और इन प्रसंस्करण कार्यों के बीच डेटा प्रवाह को ग्राफिक रूप से दर्शाया जाता है।

विश्लेषण मॉडल को इन तीन प्राथमिक उद्देश्यों को प्राप्त करना चाहिए:

वर्णन करें कि ग्राहक को क्या चाहिए।

सॉफ्टवेयर डिजाइन के निर्माण के लिए एक आधार स्थापित करें।

आवश्यकताओं के एक सेट को परिभाषित करें जिसे सॉफ्टवेयर के पूरा होने पर मान्य किया जा सकता है।

संरचित विश्लेषण तकनीक निम्नलिखित आवश्यक अंतर्निहित सिद्धांतों पर आधारित है:

ऊपर-नीचे अपघटन दृष्टिकोण

फूट डालो और जीतो सिद्धांत, जहां प्रत्येक कार्य स्वतंत्र रूप से विघटित होता है

विश्लेषण परिणामों का चित्रमय प्रतिनिधित्व

सिस्टम की कार्यक्षमता का वर्णन और विभाजन करने के लिए विशेष नोटेशन और ग्राफिकल प्रतीकों का उपयोग किया जाता है।

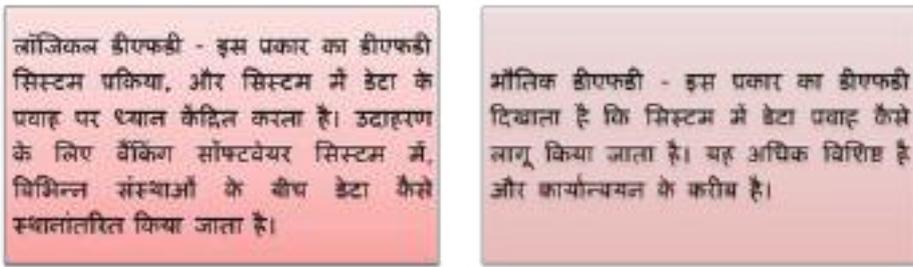
संरचित विश्लेषण के लिए निम्नलिखित उपकरणों का उपयोग किया जाता है:

डेटा प्रवाह आरेख (DFD)	संरचना चार्ट	रूपों आरेख
संरचित अंबेजी	छद्म कोड	निर्णय तालिका
ई.आर. आरेख	डेटा शब्दकोश	प्रसंग आरेख
राज्य संक्रमण आरेख	फ्लोचार्ट	घटना सूचियाँ

चित्र 4.8 संरचित विश्लेषण के लिए प्रयुक्त उपकरण

#### आंकड़ा प्रवाह आरेख

डेटा फ्लो डायग्राम (DFD) एक कंप्यूटर सिस्टम में डेटा फ्लो का ग्राफिकल प्रतिनिधित्व है। यह आने वाले डेटा प्रवाह, आउटगोइंग डेटा प्रवाह, और संग्रहीत डेटा प्रदर्शित कर सकता है। DFD इस बात का कोई उल्लेख नहीं करता है कि सिस्टम के माध्यम से डेटा कैसे प्रवाहित होता है। डेटा प्रवाह आरेख और फ्लोचार्ट के बीच एक महत्वपूर्ण अंतर है। फ्लोचार्ट प्रोग्राम मॉड्यूल के भीतर नियंत्रण प्रवाह को दर्शाता है, जबकि डीएफडी विभिन्न स्तरों पर एक सिस्टम में डेटा प्रवाह को दर्शाता है। DFD में कोई नियंत्रण या शाखा तत्व नहीं होता है। डेटा प्रवाह आरेख या तो तार्किक या भौतिक होते हैं।



चित्र 4.9 DFD के प्रकार

**डीएफडी घटक**

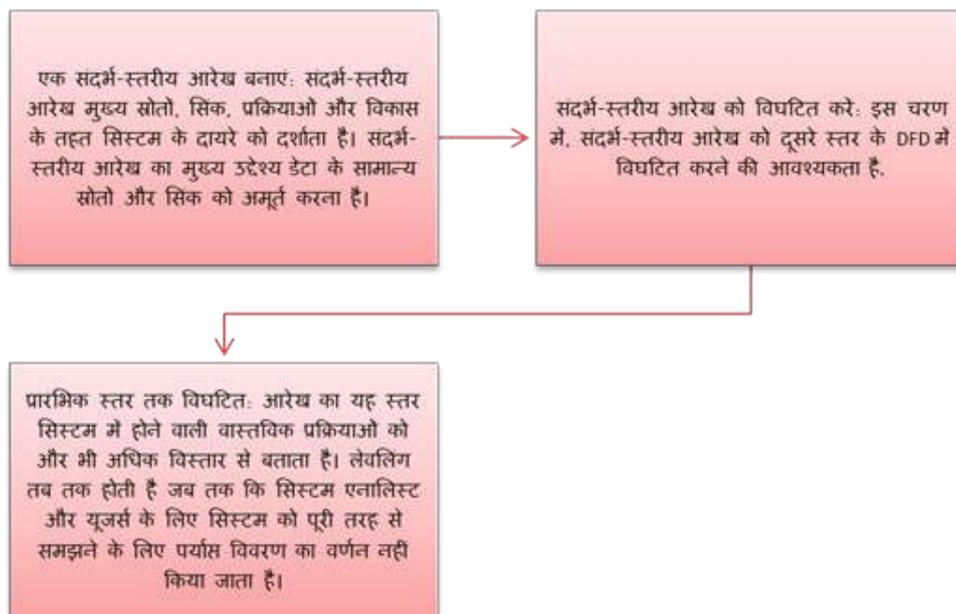
डीएफडी निम्नलिखित घटकों के सेट का उपयोग करके डेटा के स्रोत, गंतव्य, भंडारण और प्रवाह का प्रतिनिधित्व कर सकता है।



चित्र 4.10 DFD घटक

**DFD विकसित करने के चरण**

एक डीएफडी निम्नलिखित चरणों द्वारा विकसित किया जाता है:

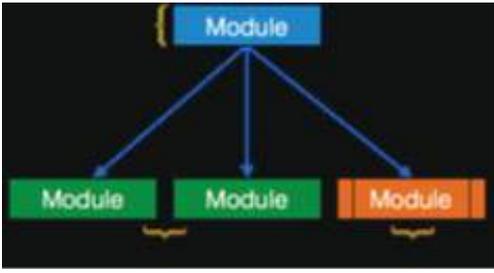
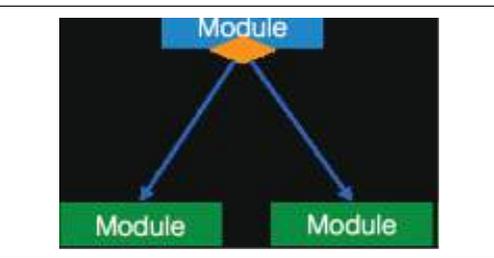
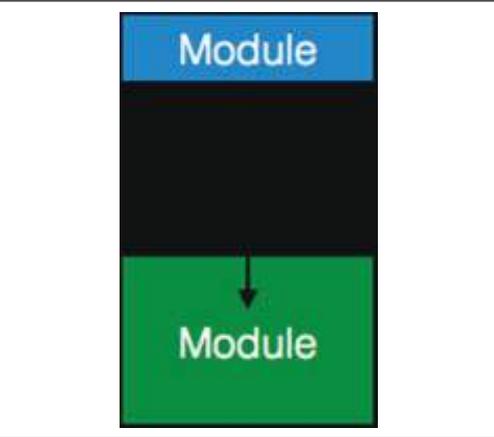
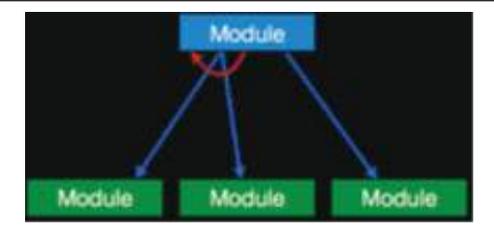


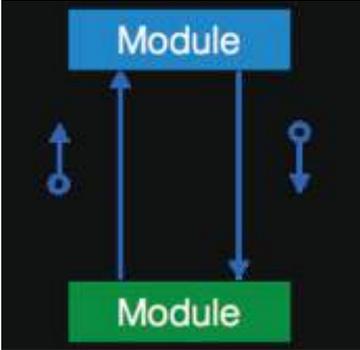
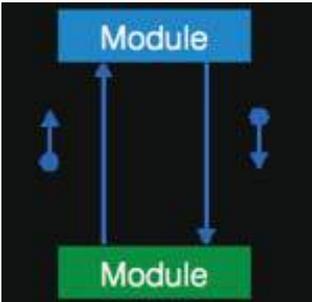
चित्र 4.11 DFD विकसित करने के चरण

### संरचना चार्ट

एक संरचना चार्ट एक प्रकार का चार्ट होता है जो डेटा प्रवाह आरेख से प्राप्त होता है। यह डीएफडी की तुलना में सिस्टम का अधिक विस्तृत प्रतिनिधित्व है। यह पूरे सिस्टम को सबसे बुनियादी कार्यात्मक मॉड्यूल में बदल देता है और डीएफडी की तुलना में प्रत्येक मॉड्यूल के कार्यों और उप-कार्यों का अधिक विस्तार से वर्णन करता है। संरचना चार्ट मॉड्यूल की पदानुक्रमित संरचना को दर्शाता है।

प्रत्येक परत पर एक विशिष्ट कार्य किया जाता है। निम्न तालिका संरचना चार्ट के निर्माण में प्रयुक्त प्रतीकों को प्रदर्शित करती है:

	<p>मॉड्यूल - यह प्रक्रिया या सबरूटीन या कार्य का प्रतिनिधित्व करता है। एक नियंत्रण मॉड्यूल एक से अधिक उप-मॉड्यूल में शाखाएं करता है। लाइब्रेरी मॉड्यूल किसी भी मॉड्यूल से पुनः उपयोग योग्य और इनवोकेबल हैं।</p>
	<p>शर्त - यह मॉड्यूल के आधार पर एक छोटे हीरे द्वारा दर्शाया गया है। यह दर्शाता है कि नियंत्रण मॉड्यूल किसी शर्त के आधार पर किसी भी उप-दिनचर्या का चयन कर सकता है।</p>
	<p>कूदना (जम्प) - मॉड्यूल के अंदर इंगित करते हुए एक तीर दिखाया गया है कि नियंत्रण कूद जाएगा उप-मॉड्यूल के बीच में।</p>
	<p>लूप - एक घुमावदार तीर मॉड्यूल में एक लूप का प्रतिनिधित्व करता है। सभी उप-मॉड्यूल, मॉड्यूल के लूप रिपीट निष्पादन द्वारा कवर किए जाते हैं।</p>

	<p>डेटा प्रवाह - अंत में एक खाली सर्कल के साथ एक निर्देशित तीर डेटा प्रवाह का प्रतिनिधित्व करता है।</p>
	<p>नियंत्रण प्रवाह - अंत में एक भरे हुए सर्कल के साथ एक निर्देशित तीर नियंत्रण प्रवाह का प्रतिनिधित्व करता है।</p>

सूची 4.1 स्ट्रक्चरल चार्ट के लिए प्रतीक

हिपो (HIPO) आरेख

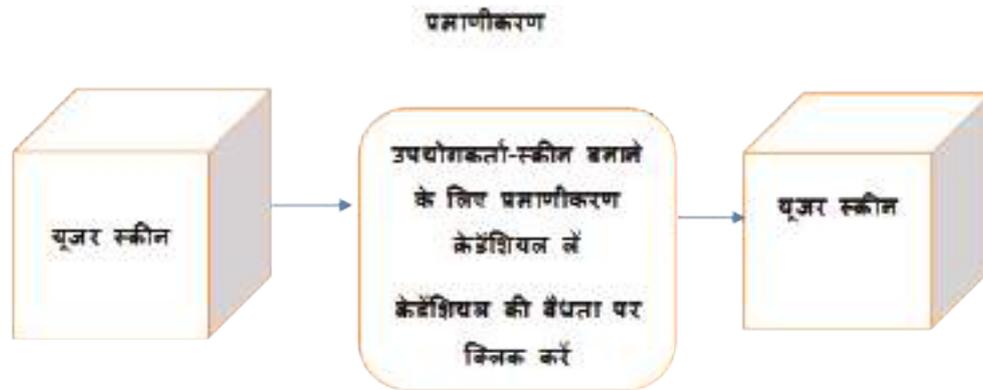
HIPO (पदानुक्रमित इनपुट प्रक्रिया आउटपुट) आरेख प्रणाली विश्लेषण और प्रलेखन के लिए दो संगठित विधियों का एक संयोजन है। वर्ष 1970 में IBM ने HIPO मॉडल बनाया।

एचआईपीओ आरेख सॉफ्टवेयर सिस्टम के मॉड्यूल पदानुक्रम को दर्शाता है। HIPO आरेख का उपयोग विश्लेषक द्वारा सिस्टम फंक्शंस का उच्च-स्तरीय दृश्य प्राप्त करने के लिए किया जाता है। यह कार्यों को उप-कार्यों में श्रेणीबद्ध रूप से विघटित करता है। यह उन कार्यों को दर्शाता है जो सिस्टम करता है। HIPO आरेख प्रलेखन के लिए उपयोगी होते हैं। उनका ग्राफिकल प्रतिनिधित्व डिजाइनरों और प्रबंधकों को सिस्टम संरचना का एक दृश्य प्रतिनिधित्व प्राप्त करने की अनुमति देता है



चित्र 4.12 HIPO आरेख

आईपीओ (इनपुट प्रोसेस आउटपुट) आरेख के विपरीत, जो एक मॉड्यूल में नियंत्रण और डेटा के प्रवाह को दर्शाता है, एचआईपीओ कोई डेटा या नियंत्रण प्रवाह जानकारी प्रदान नहीं करता है।



चित्र 4.10 इनपुट प्रक्रिया आउटपुट

#### उदाहरण

एचआईपीओ आरेख के दोनों भाग, पदानुक्रमित प्रस्तुति, और आईपीओ चार्ट का उपयोग सॉफ्टवेयर प्रोग्राम के संरचनात्मक डिजाइन के साथ-साथ उसी के प्रलेखन के लिए किया जाता है।

#### संरचित अंग्रेजी

अधिकांश प्रोग्रामर सॉफ्टवेयर की बड़ी तस्वीर से बेखबर हैं और पूरी तरह से इस बात पर भरोसा करते हैं कि उनके प्रबंधक उन्हें क्या करने के लिए कहते हैं। उच्च सॉफ्टवेयर प्रबंधन प्रोग्रामर को सटीक लेकिन तेज कोड विकसित करने के लिए सटीक जानकारी प्रदान करने के लिए जिम्मेदार है। ग्राफ या आरेखों को नियोजित करने वाली अन्य विधियों की अलग-अलग लोगों द्वारा अलग-अलग व्याख्या की जा सकती है। नतीजतन, सॉफ्टवेयर विश्लेषक और डिजाइनर संरचित अंग्रेजी जैसे उपकरण बनाते हैं। यह एक विवरण के अलावा और कुछ नहीं है कि क्या कोड किया जाना चाहिए और इसे कैसे कोडित किया जाना चाहिए। स्ट्रक्चर्ड इंग्लिश प्रोग्रामर्स को एरर-फ्री कोड लिखने में मदद करती है। ग्राफ या आरेखों को नियोजित करने वाली अन्य विधियों की अलग-अलग लोगों द्वारा अलग-अलग व्याख्या की जा सकती है। संरचित अंग्रेजी और छद्म कोड दोनों इस समझ के अंतर को पाटने का प्रयास करते हैं।

संरचित अंग्रेजी एक संरचित प्रोग्रामिंग प्रतिमान है जिसमें सादे अंग्रेजी शब्दों का उपयोग होता है।

यह अंतिम कोड नहीं है, बल्कि इसका विवरण है कि क्या कोड किया जाना चाहिए और इसे कैसे कोडित किया जाना चाहिए।

यहां संरचित प्रोग्रामिंग के कुछ उदाहरण दिए गए हैं।

अगर तब या,

करें-जबकि-तब तक

कृपया ध्यान दें कि विश्लेषक उसी चर और डेटा नाम का उपयोग करता है, जो डेटा डिक्शनरी में संग्रहीत होता है, जिससे कोड को लिखना और समझना बहुत आसान हो जाता है।

#### उदाहरण

आइए ऑनलाइन शॉपिंग वातावरण में ग्राहक प्रमाणीकरण का एक ही उदाहरण लें। ग्राहकों को प्रमाणित करने की प्रक्रिया

को संरचित अंग्रेजी में इस प्रकार लिखा जा सकता है:

ग्राहक\_नाम दर्ज करें

ग्राहक नाम \_DB फाइल में ग्राहक का नाम देखें

अगर ग्राहक नाम मिल जाता है तब

कॉल प्रक्रिया USER\_PASSWORD\_AUTHENTICATE ()

अथवा

प्रिंट त्रुटि संदेश

कॉल प्रक्रिया NEW\_CUSTOMER\_REQUEST ()

अथवा अंत

संरचित अंग्रेजी में लिखा गया कोड दिन-प्रतिदिन बोली जाने वाली अंग्रेजी की तरह है। इसे सीधे सॉफ्टवेयर के कोड के रूप में लागू नहीं किया जा सकता है। संरचित अंग्रेजी प्रोग्रामिंग भाषा से स्वतंत्र है।

स्यूडोकोड

स्यूडोकोड उसी भाषा में लिखा जाता है जिसमें प्रोग्रामिंग भाषा होती है। इसे बहुत सारी टिप्पणियों और विवरणों के साथ एक संवर्धित प्रोग्रामिंग भाषा के रूप में माना जा सकता है।

स्यूडोकोड परिवर्तनीय घोषणा से बचा जाता है, लेकिन यह वास्तविक प्रोग्रामिंग भाषा संरचनाओं जैसे सी, फोरट्रान, पास्कल, आदि का उपयोग करके लिखा जाता है।

संरचित अंग्रेजी में सूडोकोड की तुलना में कम प्रोग्रामिंग विवरण होते हैं। यह कार्य करने के लिए एक विधि प्रदान करता है जैसे कि कंप्यूटर द्वारा कोड निष्पादित किया जा रहा था।

उदाहरण

फाइबोनैचि को n संख्या तक प्रिंट करने का कार्यक्रम।

void function Fibonacci

Get the value of n;

Set value of a to 1;

Set value of b to 1;

Initialize l to 0

for (i=0; i< n; i++)

{

if a greater than b

{

Increase b by a;

Print b;

}

else if b greater than a

```
{
increase a by b;
print a;
}
}
```

फायदे
<ul style="list-style-type: none"> <li>• फ्लोचार्ट या निर्णय तालिका को परिवर्तित करने की तुलना में छद्म कोड को प्रोग्रामिंग भाषा में परिवर्तित करना बहुत आसान है।</li> <li>• फ्लोचार्ट की तुलना में, जब भी प्रोग्राम संशोधनों की आवश्यकता होती है, प्रोग्राम लॉजिक के छद्म कोड को संशोधित करना आसान होता है।</li> <li>• स्यूडोकोड लिखने में समकक्ष फ्लोचार्ट की तुलना में बहुत कम समय और प्रयास शामिल होता है।</li> <li>• एक प्रोग्रामिंग भाषा में प्रोग्राम लिखने की तुलना में छद्म कोड लिखना आसान है क्योंकि एक विधि के रूप में छद्म कोड का पालन करने के लिए केवल कुछ नियम हैं।</li> </ul>

नुकसान
<ul style="list-style-type: none"> <li>• स्यूडोकोडके मामले में, प्रोग्राम लॉजिक का ग्राफिक प्रतिनिधित्व फ्लोचार्ट के साथ उपलब्ध नहीं है।</li> <li>• स्यूडोकोडका उपयोग करने में पालन करने के लिए कोई मानक नियम नहीं है। विभिन्न प्रोग्रामर छद्म कोड लिखने की अपनी शैली का उपयोग करते हैं और इसलिए सघन</li> <li>• मानकीकरण की कमी के कारण समस्याएं होती हैं।</li> <li>• एक शुरुआत के लिए, फ्लोचार्टिंग की तुलना में तर्क का पालन करना या छद्म कोड लिखना अधिक कठिन होता है।</li> </ul>

चित्र . 4.12 स्यूडोकोड के फायदे एवं नुकसान

### निर्णय तालिका

एक निर्णय तालिका एक संरचित सारणीबद्ध प्रारूप है जो शर्तों और उन कार्यों का प्रतिनिधित्व करता है जो उन्हें संबोधित करने के लिए किए जाने चाहिए। यह एक प्रभावी डिबगिंग और त्रुटि-निवारण उपकरण है। यह समान जानकारी को एक तालिका में समूहित करने में सहायता करता है, और फिर तालिकाओं का संयोजन सरल और सुविधाजनक निर्णय लेने की सुविधा प्रदान करता है। निर्णय तालिकाएँ जटिल तर्क के प्रतिरूपण की एक सटीक लेकिन सघन विधि हैं। फ्लोचार्ट और अगर-तब-और और स्विच-केस स्टेटमेंट जैसी निर्णय तालिकाएँ, कार्रवाई के साथ शर्तों को जोड़ती हैं, लेकिन कई मामलों में अधिक सुरुचिपूर्ण तरीके से।

एक निर्णय तालिका एक तालिका होती है जिसमें पंक्तियों और स्तंभों को चार चतुर्भुजों में विभाजित किया जाता है।

स्थितियाँ शर्त विकल्प

कार्रवाई कार्रवाई प्रविष्टियाँ

स्थितियाँ	शर्त विकल्प
कार्रवाई	कार्रवाई प्रविष्टियाँ

चित्र 4.13 निर्णय तालिका चार चतुर्भुजों में विभाजित

स्थितियाँ ऊपरी बाएं चतुर्थांश में हैं। विकल्पों के शर्त नियम ऊपरी दाएँ चतुर्थांश में स्थित हैं।

की जाने वाली क्रियाएँ निचले बाएं चतुर्थांश में हैं, और क्रिया नियम निचले दाएँ चतुर्थांश में हैं। निर्णय तालिकाएँ जटिल तर्क के प्रतिरूपण की एक सटीक लेकिन सघन विधि हैं।

निर्णय तालिकाएँ, जैसे कि अगर-तब-और और स्विच-केस स्टेटमेंट, क्रियाओं के लिए शर्तों को लिंक करती हैं। हालांकि, पारंपरिक प्रोग्रामिंग भाषा नियंत्रण संरचनाओं के विपरीत, निर्णय तालिकाएँ कई स्वतंत्र स्थितियों को कई क्रियाओं के

साथ जोड़ सकती हैं।

निर्णय तालिका बनाना

निर्णय तालिका बनाने के लिए, डेवलपर को बुनियादी चार चरणों का पालन करना होगा:



चित्र 4.14 निर्णय तालिका बनाने के चरण

निर्णय तालिका को अंतिम उपयोगकर्ताओं द्वारा सत्यापित किया जाना चाहिए और हाल ही में डुप्लिकेट नियमों और कार्यों को समाप्त करके इसे सरल बनाया जा सकता है।

निर्णय तालिका विकसित करने के लिए कदम: स्पष्टीकरण

निर्णय तालिकाएँ बनाने के लिए, आपको तालिका के अधिकतम आकार को निर्धारित करने, किसी भी असंभव स्थितियों, विसंगतियों या अतिरिक्त को समाप्त करने और तालिका को यथासंभव सरल बनाने की आवश्यकता है। निम्नलिखित कदम निर्णय तालिका विकसित करने के लिए कुछ दिशानिर्देश प्रदान करते हैं:

निर्णय को प्रभावित करने वाली स्थितियों की संख्या निर्धारित करें। ओवरलैप करने वाली पंक्तियों को मिलाएं, उदाहरण के लिए, परस्पर अनन्य स्थितियां। शर्तों की संख्या निर्णय तालिका के शीर्ष भाग में पंक्तियों की संख्या बन जाती है।

की जा सकने वाली संभावित कार्रवाइयों की संख्या निर्धारित करें। यह निर्णय तालिका के निचले आधे भाग में पंक्तियों की संख्या बन जाती है।

द्वितीय. प्रत्येक शर्त के लिए शर्त विकल्पों की संख्या निर्धारित करें। निर्णय तालिका के सरलतम रूप में, प्रत्येक शर्त के लिए दो विकल्प (Y या N) होंगे। एक विस्तारित-प्रविष्टि तालिका में, प्रत्येक शर्त के लिए कई विकल्प हो सकते हैं।

प्रत्येक शर्त के लिए विकल्पों की संख्या को गुणा करके निर्णय तालिका में स्तंभों की अधिकतम संख्या की गणना करें। यदि प्रत्येक शर्त के लिए चार शर्त और दो विकल्प (Y या N) हों, तो सोलह संभावनाएं इस प्रकार होंगी:

शर्त 1: x 2 विकल्प

शर्त 2: x 2 विकल्प

शर्त 3: x 2 विकल्प

शर्त 4: x 2 विकल्प

16 संभावनाएं

iv शर्त विकल्प भरें। पहली शर्त से शुरू करें और उस स्थिति के लिए विकल्पों की संख्या से कॉलम की संख्या को विभाजित करें। पूर्वगामी उदाहरण में, सोलह कॉलम और दो विकल्प (Y और N) हैं, इसलिए सोलह को दो से विभाजित करने पर आठ होता है। फिर विकल्पों में से किसी एक को चुनें और सभी आठ कॉलमों में Y लिखें। शेष आठ कॉलमों में इस प्रकार N लिखकर समाप्त करें:

शर्त 1 YYYYYYYNNNNNNNN

तालिका के सबसेट का उपयोग करके प्रत्येक शर्त के लिए इसे दोहराएं

Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
N	N	N	N	N						
Y	Y	Y	Y	N	N	N	N			
Y	Y	N	N							
Y	N									

सूची 4.2 तालिका के सबसेट

प्रत्येक शर्त के लिए पैटर्न जारी रखें:

शर्त 1	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
शर्त 2	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
शर्त 3	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
शर्त 4	Y	N	Y	N	Y	N										

सूची 4.3 (a) प्रत्येक शर्त के लिए तालिका

V. एक X डालकर तालिका को पूरा करें जहां नियम कुछ क्रियाओं का सुझाव देते हैं।

उदाहरण के लिए:

शर्त 1 Y Y

शर्त 2 Y N

क्रिया 1 X X

के रूप में व्यक्त किया जा सकता है:

शर्त 1 Y

शर्त 2 --

क्रिया 1 X

डैश (-) का अर्थ है कि शर्त 2 या तो Y या N हो सकती है और फिर भी कार्रवाई की जाएगी।

VI. किसी भी असंभव स्थितियों, विरोधाभासों या अतिरेक के लिए तालिका देखें।

9. निर्णय तालिका को और अधिक समझने योग्य बनाने के लिए शर्तों और कार्यों (या यहां तक कि नियम) को पुनर्व्यवस्थित करें।

उदाहरण: जब तक कटौती योग्य राशि पूरी नहीं हो जाती, तब तक रोगी को किसी शुल्क की प्रतिपूर्ति नहीं की जाती है। कटौती योग्य मिलने के बाद, प्रतिपूर्ति की जाने वाली राशि इस बात पर निर्भर करती है कि डॉक्टर या अस्पताल "पसंदीदा प्रदाता" है या नहीं। पसंदीदा प्रदाताओं के लिए, डॉक्टर के कार्यालय के दौरे की प्रतिपूर्ति 65% पर की जाती है और अस्पताल के दौरे की 95% पर प्रतिपूर्ति की जाती है। अन्य प्रदाताओं के लिए डॉक्टर के कार्यालय के दौरे के लिए 50% या अस्पताल के दौरे के लिए 80% की प्रतिपूर्ति करें।

8 नियम होंगे। पहली शर्त (क्या कटौती योग्य है?) के दो संभावित परिणाम हैं, हां या नहीं। दूसरी शर्त (क्या यह एक पसंदीदा प्रदाता है?) के दो संभावित परिणाम हैं, हां या नहीं। तीसरी स्थिति (यात्रा का प्रकार) के दो संभावित परिणाम हैं, डॉक्टर के कार्यालय का दौरा (डी) या अस्पताल का दौरा (एच)। दो गुना दो गुना दो है 8.

शर्त 1	1	2	3	4	5	6	4	8
कटौती मिली?	Y	Y	Y	Y	N	N	N	N
पसंदीदा प्रदाता?	Y	Y	N	N	Y	Y	N	N
यात्रा का प्रकार कार्यवाही	D	H	D	H	D	H	D	H
65% भरपाई	X							
95% भरपाई		X						
50% भरपाई			X					
80% भरपाई				X				
कोई भरपाई नहीं					X	X	X	X

सूची 4.4 (b) प्रत्येक शर्त की तालिका

उदाहरण

आइए हम अपनी इंटरनेट कनेक्टिविटी के साथ दिन-प्रतिदिन की समस्या का एक सरल उदाहरण लेते हैं। हम उन सभी समस्याओं की पहचान करके शुरू करते हैं जो इंटरनेट और उनके संबंधित संभावित समाधानों को शुरू करते समय उत्पन्न हो सकती हैं।

हम कॉलम शर्तों के तहत सभी संभावित समस्याओं और कॉलम क्रियाओं के तहत संभावित क्रियाओं को सूचीबद्ध करते हैं।

शर्त	शर्त /कार्यवाही	नियम							
		N	N	N	N	N	N	N	N
शर्त	जुड़ा हुआ दिखाता है	N	N	N	N	N	N	N	N
	पिंग काम कर रहा है	N	N	Y	Y	N	N	Y	Y
	वेबसाइट खोलता है	Y	N	Y	N	Y	N	Y	N
कार्यवाही	नेटवर्क केबल की जाँच करें	X							
	इंटरनेट राउटर की जाँच करें	X				X	X	X	
	वेब ब्राउज़र को पुनरारंभ करें							X	
	संपर्क सेवा प्रदाता		X	X	X	X	X	X	
	कोई कार्रवाई न करें								

सूची 4.4 निर्णय तालिका-इन-हाउस इंटरनेट समस्या निवारण

निर्णय तालिका के लाभ

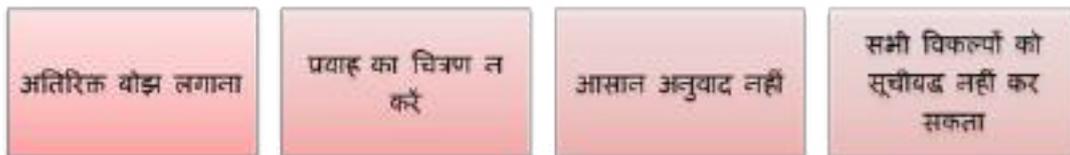
निर्णय तालिका के विभिन्न लाभों में शामिल हैं:



चित्र 4.15 निर्णय तालिका के लाभ

निर्णय तालिका के नुकसान

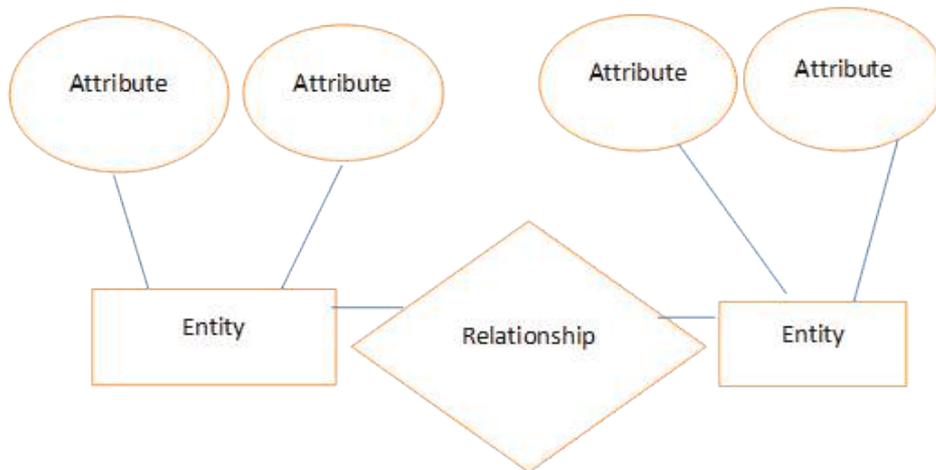
निर्णय तालिका के विभिन्न नुकसानों में शामिल हैं:



चित्र 4.16 निर्णय तालिका के नुकसान

#### 4. इकाई-संबंध मॉडल

एंटीटी-रिलेशनशिप (ईआर) मॉडल एक प्रकार का डेटाबेस मॉडल है जो वास्तविक दुनिया की संस्थाओं और उनके संबंधों की अवधारणा पर आधारित है। वास्तविक दुनिया के परिदृश्यों को ईआर डेटाबेस मॉडल पर मैप किया जा सकता है। ईआर मॉडल विशेषताओं, बाधाओं और उनके बीच संबंधों के साथ संस्थाओं का एक सेट उत्पन्न करता है। ईआर मॉडल डेटाबेस वैचारिक डिजाइन के लिए सबसे उपयुक्त है। ईआर मॉडल निम्नानुसार दर्शाया गया है:



चित्र 4.17 (a) निकाय सम्बंधित मॉडल

संबंधों को और अधिक परिभाषित करने के लिए ईआर आरेख।

निकाय: एक निकाय एक भौतिक वस्तु या अवधारणा है जिसके बारे में आप डेटा संग्रहीत करना चाहते हैं। यह एक व्यक्ति, स्थान, घटना या वस्तु हो सकती है जो किसी विशेष प्रणाली के लिए महत्वपूर्ण है। उदाहरण के लिए, एक स्कूल प्रणाली में छात्र, शिक्षक, प्रमुख पाठ्यक्रम, विषय, शुल्क और अन्य आइटम शामिल हो सकते हैं। ईआर आरेखों में, संस्थाओं को एक आयत द्वारा दर्शाया जाता है और एकवचन संज्ञाओं के साथ नामित किया जाता है।



चित्र 4.17 (b) निकाय सम्बंधित मॉडल

कमजोर निकाय : एक कमजोर निकाय एक निकाय है जिसे किसी अन्य निकाय के साथ एक विदेशी कुंजी संबंध द्वारा परिभाषित किया जाना चाहिए क्योंकि इसे केवल अपने स्वयं के गुणों द्वारा विशिष्ट रूप से पहचाना नहीं जा सकता है।



चित्र 4.17 (c) निकाय सम्बंधित मॉडल

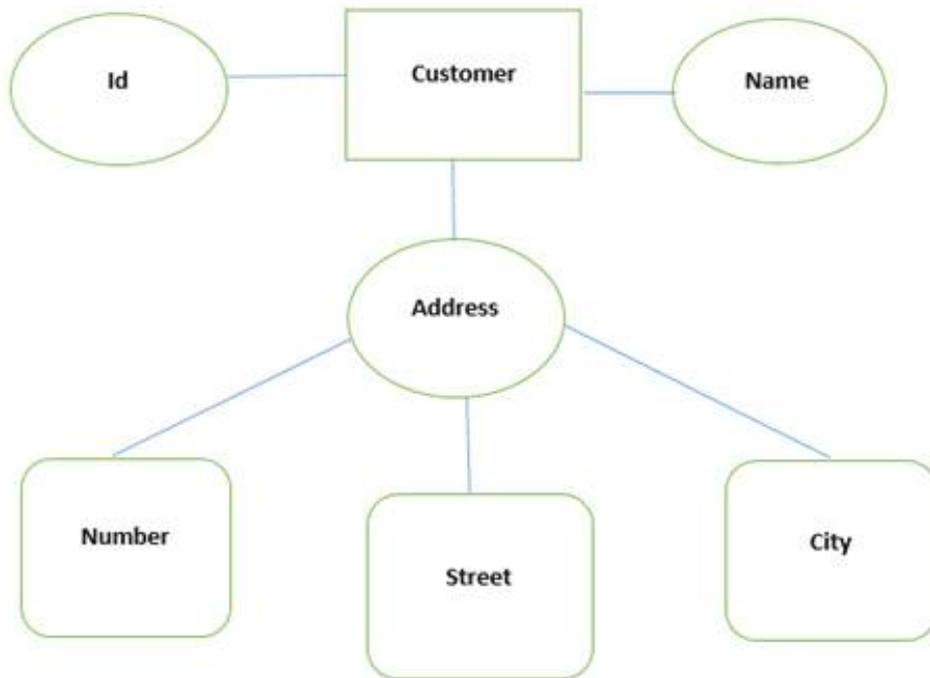
एक कमजोर निकाय एक निकाय है जो किसी अन्य इकाई के अस्तित्व पर निर्भर करती है। अधिक तकनीकी शब्दों में इसे एक ऐसी इकाई के रूप में परिभाषित किया जा सकता है जिसे अपनी विशेषताओं से पहचाना नहीं जा सकता है। यह के साथ संयुक्त एक विदेशी कुंजी का उपयोग करता है

इसे प्राथमिक कुंजी बनाने के लिए जिम्मेदार ठहराया गया है। ऑर्डर आइटम जैसी इकाई इसके लिए एक अच्छा उदाहरण है। ऑर्डर आइटम ऑर्डर के बिना अर्थहीन होगा इसलिए यह ऑर्डर के अस्तित्व पर निर्भर करता है।

**Order Item**

चित्र 4.17 (d) निकाय सम्बंधित मॉडल

o विशेषता: एक विशेषता एक संपत्ति, विशेषता, या एक इकाई, संबंध, या किसी अन्य विशेषता की विशेषता है। उदाहरण के लिए, विशेषता वस्तु सूची आइटम नाम इकाई वस्तु सूची आइटम की एक विशेषता है। एक इकाई में जितनी आवश्यकता हो उतने गुण हो सकते हैं। इस बीच, विशेषताओं की अपनी विशिष्ट विशेषताएँ भी हो सकती हैं। उदाहरण के लिए, विशेषता "ग्राहक पता" में विशेषता संख्या, सड़क, शहर और राज्य हो सकता है। इन्हें संयुक्त गुण कहते हैं। ध्यान दें कि कुछ शीर्ष स्तर के ईआर आरेख सरलता के लिए विशेषताएँ नहीं दिखाते हैं। हालांकि, ऐसा करने वालों में, विशेषताओं को अंडाकार आकृतियों द्वारा दर्शाया जाता है।



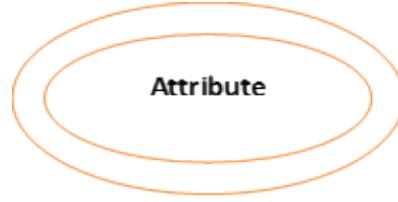
चित्र 4.18 विशेषता -निकाय सम्बंधित मॉडल

मुख्य विशेषता: एक प्रमुख विशेषता इकाई की अनूठी, विशिष्ट विशेषता है। उदाहरण के लिए, किसी कर्मचारी की सामाजिक सुरक्षा संख्या कर्मचारी की प्रमुख विशेषता हो सकती है।

**Attribute**

चित्र 4.18 (a) विशेषता -निकाय सम्बंधित मॉडल

बहु-मूल्यवान विशेषता: एक बहु-मूल्यवान विशेषता में एक से अधिक मान हो सकते हैं। उदाहरण के लिए, एक कर्मचारी इकाई में कई कौशल मूल्य हो सकते हैं।



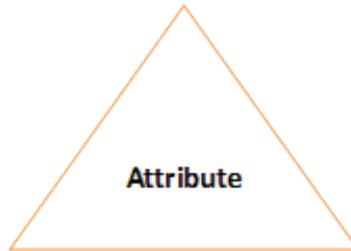
चित्र 4.18 (b) बहु-मूल्यवान विशेषता -निकाय सम्बंधित मॉडल

यह ध्यान रखना महत्वपूर्ण है कि यह एक विशेषता के अपने गुणों से अलग है। उदाहरण के लिए, एक शिक्षक इकाई में कई विषय मान हो सकते हैं।



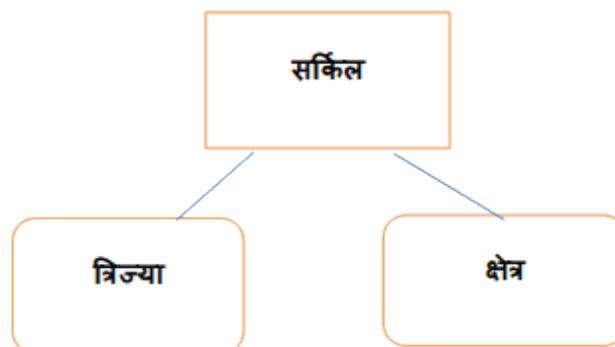
चित्र 4.18 (c) बहु-मूल्यवान विशेषता - इकाई-संबंध मॉडल

व्युत्पन्न विशेषता: एक व्युत्पन्न विशेषता किसी अन्य विशेषता पर आधारित होती है। उदाहरण के लिए, किसी कर्मचारी का मासिक वेतन कर्मचारी के वार्षिक वेतन पर आधारित होता है।



चित्र 4.18 (d) व्युत्पन्न विशेषता - इकाई-संबंध मॉडल

एक अन्य विशेषता पर आधारित विशेषता है। यह ईआर आरेखों में शायद ही कभी पाया जाता है। उदाहरण के लिए, एक वृत्त का क्षेत्रफल त्रिज्या से निकाला जा सकता है।



चित्र 4.18 (e) व्युत्पन्न विशेषता - इकाई-संबंध मॉडल

संबंध: संबंध बताते हैं कि कैसे दो संस्थाएं डेटाबेस संरचना में जानकारी साझा करती हैं



चित्र 4.19 संबंध - निकाय -संबंध मॉडल

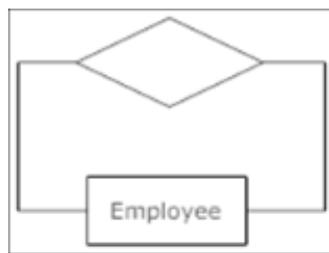
एक रिश्ता बताता है कि संस्थाएं कैसे बातचीत करती हैं। उदाहरण के लिए, इकाई "बढ़ई" का संबंध "निर्माण" या "बनाने" के संबंध में इकाई "तालिका" से हो सकता है। रिश्तों को हीरे के आकार द्वारा दर्शाया जाता है और क्रियाओं का उपयोग करके लेबल किया जाता है।

कार्डिनैलिटी: कार्डिनैलिटी बताती है कि एक इकाई के कितने उदाहरण दूसरे के एक उदाहरण से संबंधित हैं। कार्डिनैलिटी का ऑर्डिनैलिटी से भी गहरा संबंध है। यह एक रिश्ते की घटनाओं को निर्दिष्ट करता है, जबकि सामान्यता निर्दिष्ट करती है कि संबंध अनिवार्य या वैकल्पिक है या नहीं। उदाहरण के लिए, ईमेल सिस्टम में एक खाते में कई संपर्क हो सकते हैं। इस मामले में, संबंध "एक से कई" मॉडल का अनुसरण करता है। ईआर आरेखों में, विभिन्न प्रकार के नोटेशन का उपयोग करके कार्डिनैलिटी का प्रतिनिधित्व किया जाता है।



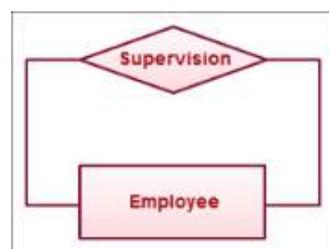
चित्र 4.20 प्रमुखता

पुनरावर्ती संबंध: कुछ मामलों में, निकाय स्व-जुड़े हो सकते हैं। उदाहरण के लिए, कर्मचारी अन्य कर्मचारियों की निगरानी कर सकते हैं।



चित्र 4.21 (a) पुनरावर्ती संबंध

नीचे दिए गए उदाहरण में एक कर्मचारी पर्यवेक्षक हो सकता है और उसकी देखरेख की जा सकती है, इसलिए एक पुनरावर्ती संबंध है।



चित्र 4.21 (b) पुनरावर्ती संबंध

### ईआर आरेखों के लाभ

ईआर डायग्राम डेटाबेस बनाने और हेरफेर करने के लिए एक उत्कृष्ट ढांचा प्रदान करते हैं।

उन्हें समझना आसान है और कुशलतापूर्वक और सटीक रूप से काम करने के लिए व्यापक प्रशिक्षण की आवश्यकता नहीं है और डिजाइनर, आईटी दक्षता की परवाह किए बिना, डेवलपर्स, ग्राहकों और अंतिम उपयोगकर्ताओं के साथ संवाद करने के लिए आसानी से ईआर आरेखों का उपयोग कर सकते हैं। ए

ईआर आरेख आसानी से संबंधपरक तालिकाओं में अनुवाद योग्य होते हैं, जिनका उपयोग जल्दी से डेटाबेस बनाने के लिए किया जा सकता है।

इसके अलावा, डेटाबेस डेवलपर्स विशिष्ट सॉफ्टवेयर अनुप्रयोगों में डेटा को लागू करने के लिए ब्लूप्रिंट के रूप में ईआर डायग्राम का उपयोग कर सकते हैं।

अंत में, किसी संगठन के भीतर विभिन्न संबंधों और संचालन का वर्णन करने के लिए ईआर आरेखों का उपयोग किया जा सकता है।

### डेटा शब्दकोश

डेटा डिक्शनरी डेटा से संबंधित जानकारी का एक केंद्रीकृत संग्रह है। यह डेटा अर्थ और मूल, अन्य डेटा के साथ संबंध, उपयोग के लिए डेटा प्रारूप, और इसी तरह बचाता है। डेटा डिक्शनरी के सभी नामों में उपयोगकर्ताओं और सॉफ्टवेयर डिजाइनरों की मदद करने के लिए कठोर परिभाषाएँ हैं। डीएफडी (डेटा फ्लो डायग्राम) सॉफ्टवेयर प्रोग्राम मॉडल जिसे जब भी डीएफडी को बदला या अपडेट किया जाता है तो अपडेट होने की उम्मीद है।

### डेटा डिक्शनरी की आवश्यकता

सॉफ्टवेयर को डिजाइन और कार्यान्वित करते समय, डेटा को डेटा डिक्शनरी का उपयोग करके संदर्भित किया जाता है। एक डेटा डिक्शनरी अस्पष्टता की संभावना को समाप्त करती है। यह पूरे कार्यक्रम में एक ही वस्तु संदर्भ का उपयोग करके प्रोग्रामर और डिजाइनरों के काम को सिंक्रनाइज़ रखने में सहायता करता है। डेटा डिक्शनरी संपूर्ण डेटाबेस सिस्टम के लिए केंद्रीकृत दस्तावेज़ीकरण प्रदान करता है। डेटा डिक्शनरी का उपयोग DFD को मान्य करने के लिए किया जाता है।

### अंतर्वस्तु

निम्नलिखित जानकारी को डेटा डिक्शनरी में शामिल किया जाना चाहिए:



चित्र 4.22 डेटा शब्दकोश

डेटा प्रवाह को डीएफडी के रूप में निर्दिष्ट किया गया है जैसा कि पहले अध्ययन किया गया था और वर्णित के रूप में बीजगणितीय रूप में दर्शाया गया है।

=	Composed of
{ }	Repetition
{ }	Optional
+	And
[/]	OR

चित्र 4.23 बीजीय रूप में DFD

#### उदाहरण

पता = मकान नंबर + (सड़क / क्षेत्र) + शहर + राज्य

कोर्स आईडी = कोर्स नंबर + कोर्स का नाम + कोर्स लेवल + कोर्स ग्रेड

डेटा तत्वों में निम्नलिखित विवरण के साथ डेटा और नियंत्रण आइटम, आंतरिक या बाहरी डेटा स्टोर आदि के नाम और विवरण शामिल हैं:

प्राथमिक नाम

माध्यमिक नाम (उपनाम)

उपयोग-मामला (कैसे और कहाँ उपयोग करें)

सामग्री विवरण (नोटेशन आदि)

अनुपूरक सूचना (वर्तमान मूल्य, बाधाएं आदि)

यह उन सूचनाओं को संग्रहीत करता है जहां से डेटा सिस्टम में प्रवेश करता है और सिस्टम से बाहर मौजूद होता है। डेटा स्टोर में शामिल हो सकते हैं -

फाइल्स

सॉफ्टवेयर के लिए आंतरिक।

सॉफ्टवेयर के लिए बाहरी लेकिन एक ही मशीन पर।

विभिन्न मशीन पर स्थित सॉफ्टवेयर और सिस्टम के लिए बाहरी।

टेबल

नामकरण परंपरा

अनुक्रमण संपत्ति

डाटा प्रासेसिंग

डाटा प्रोसेसिंग दो प्रकार की होती है:

तार्किक: जैसा कि उपयोगकर्ता इसे देखता है

भौतिक: जैसा कि सॉफ्टवेयर इसे देखता है

#### प्रसंग आरेख

स्कोप विवरण विकास के तहत प्रणाली और ब्रह्मांड में बाकी सब कुछ के बीच अंतर को परिभाषित करता है। संदर्भ आरेख इस सीमा को विकासाधीन या विचाराधीन समस्या और बाहरी दुनिया के बीच संबंधों को प्रदर्शित करके इस सीमा को दर्शाता है। संदर्भ आरेख सिस्टम के बाहर की संस्थाओं की पहचान करता है जो इसके साथ किसी तरह से इंटरैक्ट करते हैं - टर्मिनेटर या बाहरी संस्थाओं के रूप में संदर्भित, साथ ही साथ प्रत्येक बाहरी इकाई और सिस्टम के बीच डेटा और सामग्री का प्रवाह। संरचित विश्लेषण सिद्धांतों का उपयोग करके बनाए गए डेटा प्रवाह आरेख में संदर्भ आरेख का उपयोग शीर्ष-स्तरीय अमूर्तता के रूप में किया जाता है। संदर्भ आरेख को विज्ञान और स्कोप दस्तावेज़, SRS, या सिस्टम के डेटाफ्लो मॉडल के भाग के रूप में शामिल किया जा सकता है।

ऑनलाइन सामुदायिक प्रणाली का वर्णन करने वाले संदर्भ आरेख का एक नमूना निम्नलिखित है।



चित्र 4.24 ऑनलाइन सामुदायिक प्रणाली के लिए सामग्री आरेख

(Image Source: <https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/1433/What-is-a-Context-Diagram-and-what-are-the-benefits-of-creating-one.aspx>)

#### राज्य-संक्रमण आरेख

एक राज्य-संक्रमण आरेख (एसटीडी) एक प्रणाली के समय-निर्भर पहलुओं के मॉडलिंग के लिए एक उपकरण है। एक एसटीडी एक उपकरण है जिसका उपयोग कई रीयल-टाइम सिस्टम के अनुक्रम को अनुकरण करने के लिए किया जाता है। एसटीडी समय पर निर्भर व्यवहार का प्रतिनिधित्व करता है। कई जटिल प्रणालियाँ समय-निर्भर व्यवहार प्रदर्शित करती हैं, जो उस क्रम को संदर्भित करता है जिसमें डेटा तक पहुँचा जाता है और कार्य किया जाता है।

एक एसटीडी एक सिस्टम के समय-निर्भर व्यवहार का वर्णन करने का एक तरीका है। मूल संगति नियम हैं: "किसी भी राज्य में एक प्रणाली का व्यवहार समान होना चाहिए, चाहे वह राज्य किसी भी रास्ते पर क्यों न पहुँचे"।

एक राज्य प्रणाली के व्यवहार का एक अवलोकन योग्य तरीका है।

एक विशिष्ट एसटीडी किसी भी समय केवल एक ही राज्य में हो सकता है।

हालाँकि, एक सिस्टम के व्यवहार को एक से अधिक राज्य संक्रमण आरेख द्वारा वर्णित किया जा सकता है।

चित्र 4.25 राज्य-संक्रमण आरेख

संक्रमण की स्थिति:

आंतरिक घटनाएं या सिस्टम के बाहर की घटनाएं

संक्रमण क्रियाएँ:

घटनाओं के जवाब में कार्रवाई

एक-शॉट क्रियाओं को ट्रिगर करना

विभिन्न एसटीडी के बीच सिंक्रनाइज़ करना

उत्पादन नियंत्रण आउटपुट

ड्राइंग एसटीडी:

सिस्टम के अवलोकन योग्य राज्यों की पहचान करें

सामान्य व्यवहार वाले राज्यों का चयन करें

उन शर्तों को निर्दिष्ट करें जो एक संक्रमण को चिह्नित करती हैं

प्रत्येक संक्रमण के लिए गंतव्य स्थिति में देखने योग्य व्यवहार उत्पन्न करने के लिए क्रियाओं को निर्दिष्ट करें

यदि सिस्टम जटिल है, तो आरेख को कई एसटीडी में विभाजित करें

शतरंज के खेल के नियंत्रण प्रवाह को दर्शाने वाला एक राज्य संक्रमण आरेख चित्रबेलो में दर्शाया गया है। प्रारंभ को प्रारंभिक स्थिति के रूप में परिभाषित किया गया है, जबकि 'ब्लैक जीत', 'ड्रा', और 'व्हाइट जीत' सिस्टम के लिए तीन अंतिम राज्य हैं।

संक्रमण की स्थिति:

आंतरिक घटनाएं या सिस्टम के बाहर की घटनाएं

संक्रमण क्रियाएँ:

घटनाओं के जवाब में कार्रवाई

एक-शॉट क्रियाओं को ट्रिगर करना

विभिन्न एसटीडी के बीच सिंक्रनाइज़ करना

उत्पादन नियंत्रण आउटपुट

ड्राइंग एसटीडी:

सिस्टम के अवलोकन योग्य राज्यों की पहचान करें

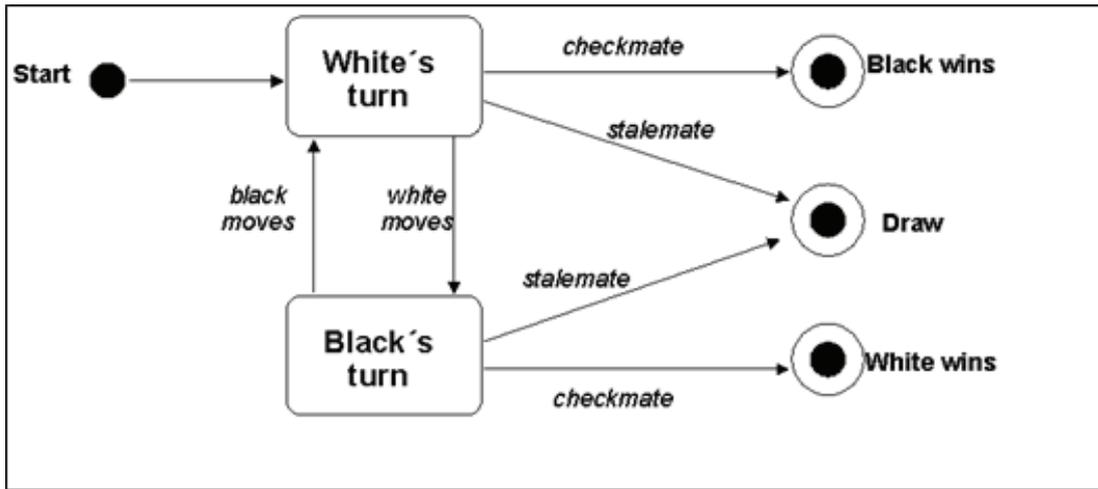
सामान्य व्यवहार वाले राज्यों का चयन करें

संक्रमण को चिह्नित करने वाली शर्तों को निर्दिष्ट करें

प्रत्येक संक्रमण के लिए गंतव्य स्थिति में देखने योग्य व्यवहार उत्पन्न करने के लिए क्रियाओं को निर्दिष्ट करें

यदि सिस्टम जटिल है, तो आरेख को कई एसटीडी में विभाजित करें

शतरंज के खेल के नियंत्रण प्रवाह को दर्शाने वाला एक राज्य संक्रमण आरेख चित्रबेलो में दर्शाया गया है। प्रारंभ को प्रारंभिक स्थिति के रूप में परिभाषित किया गया है, जबकि 'ब्लैक जीत', 'ड्रॉ', और 'व्हाइट जीत' सिस्टम के लिए तीन अंतिम राज्य हैं।



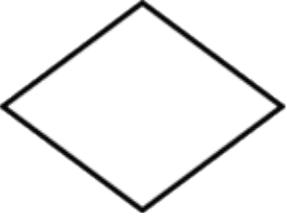
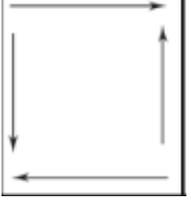
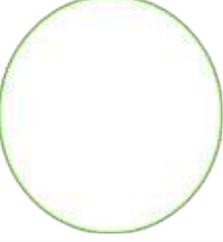
चित्र 4.26 शतरंज खेल राज्य संक्रमण आरेख - नमूना

(Image Source: <https://users.csc.calpoly.edu/~jdalbey/SWE/Design/STDexamples.html>)

### फ्लोचार्ट

एक कार्यक्रम में नियंत्रण के प्रवाह का प्रतिनिधित्व करने के लिए एक फ्लोचार्ट एक उपयोगी उपकरण है। एक फ्लोचार्ट एक एल्गोरिथम का एक चित्रमय प्रतिनिधित्व है जो किसी समस्या को हल करते समय कंप्यूटर द्वारा किए जाने वाले संचालन और निर्णयों को चित्रित करने के लिए प्रतीकों का उपयोग करता है। निर्देश स्वयं स्पष्ट कथनों का उपयोग करते हुए प्रतीकों या बक्सों के भीतर लिखे गए हैं। इन बक्सों को एक क्रम में संचालन के प्रवाह को इंगित करने वाले तीरों के साथ ठोस रेखाओं द्वारा एक साथ जोड़ा जाता है। निम्न तालिका फ्लोचार्ट प्रतीकों को प्रदर्शित करती है:

संकेत	विवरण
	टर्मिनल प्रतीक- टर्मिनल प्रतीकों का उपयोग दो उद्देश्यों के लिए किया जाता है: फ्लोचार्ट के शुरुआती (START या BEGIN) बिंदु को परिभाषित करने के लिए और फ्लोचार्ट के अंतिम बिंदु (END) को परिभाषित करने के लिए।
	इनपुट / आउटपुट प्रतीक- इनपुट/आउटपुट प्रतीकों का उपयोग इनपुट/आउटपुट संचालन की तार्किक स्थिति को इंगित करने के लिए किया जाता है। इनपुट ऑपरेशन कंप्यूटर डेटा की प्रविष्टि है और आउटपुट ऑपरेशन प्रदर्शित आउटपुट ऑपरेशन है।
	प्रसंस्करण प्रतीक- प्रसंस्करण प्रतीकों का उपयोग अंकगणित और डेटा-आंदोलन निर्देशों को इंगित करने के लिए किया जाता है। इसलिए, जोड़ने, घटाने, गुणा करने और विभाजित करने के सभी अंकगणितीय प्रसंस्करण को एक प्रसंस्करण प्रतीक बॉक्स के साथ दर्शाया जाता है।

	<p>निर्णय प्रतीक- निर्णय प्रतीकों में एक प्रवेश बिंदु होता है और प्रतीक के अंदर लिए गए निर्णय के आधार पर कम से कम दो निकास बिंदु होंगे। जब किसी शर्त का परीक्षण किया जाता है, यदि स्थिति सत्य है, तो "हां" के पथ का अनुसरण किया जाता है। यदि शर्त गलत है, तो "नहीं" के पथ का अनुसरण किया जाता है।</p>
	<p>फ्लो लाइन्स- प्रवाह रेखाएँ, जिनमें तीर के निशान होते हैं, का उपयोग फ्लोचार्ट में प्रोग्राम लॉजिक के प्रवाह को इंगित करने के लिए किया जाता है। इन तीरों का उपयोग नियंत्रण के प्रवाह की दिशा को इंगित करने के लिए किया जाता है। इसका मतलब है कि ये बयान अगले बयान को निष्पादित करने का संकेत देते हैं।</p>
	<p>कनेक्टर प्रतीक- यदि फ्लोचार्ट किसी बिंदु पर बंद कर दिया जाता है और किसी अन्य स्थान पर फिर से जारी रहता है, तो कनेक्टर प्रतीक का उपयोग किया जाता है। यह एक वृत्त है जिसके अंदर एक अंक लिखा होता है। यदि किसी बिंदु पर फ्लोचार्ट बंद कर दिया जाता है, तो चार्ट से दूर की ओर इशारा करते हुए एक वृत्त खींचा जाता है। उसी संख्या के साथ एक और सर्कल रखा गया है जहां फ्लोचार्ट जारी है।</p>
	<p>षट्भुज (सपाट)- यह तैयारी बॉक्स है। इस बॉक्स में लूप-सेटिंग स्टेटमेंट है, यानी कुछ पुनरावृत्त कथन।</p>

सूची 4.5 फ्लोचार्ट संकेत

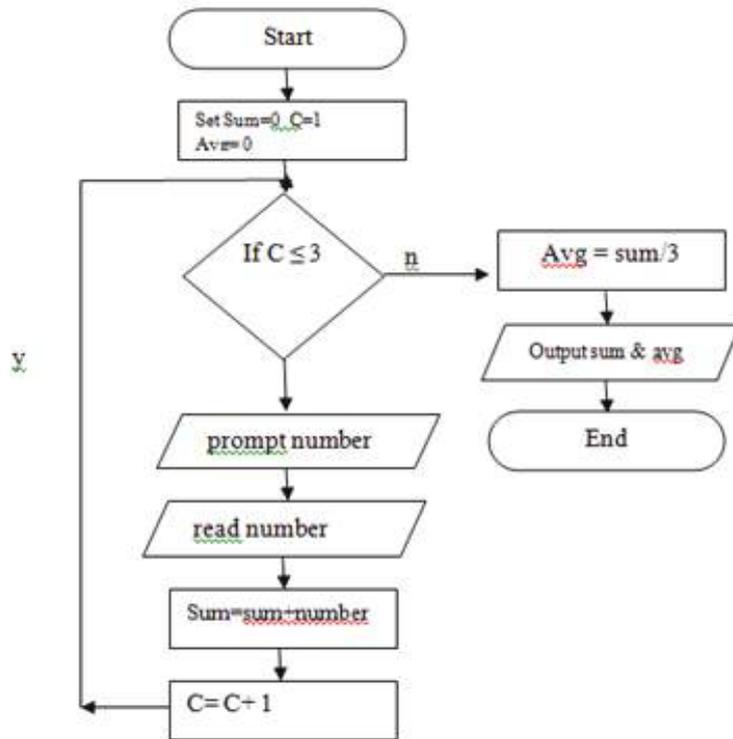
निम्नलिखित महत्वपूर्ण फ्लोचार्ट नियम और दिशानिर्देश हैं:

<p>केवल पारंपरिक फ्लोचार्ट प्रतीकों का उपयोग किया जाना चाहिए।</p>	<p>समस्या में नियंत्रण के प्रवाह को इंगित करने के लिए तीरों का उपयोग किया जा सकता है। हालांकि, प्रवाह रेखाएँ एक दूसरे को पार नहीं करनी चाहिए।</p>	<p>प्रसंस्करण तर्क ऊपर से नीचे और बाएँ से दाएँ प्रवाहित होना चाहिए।</p>
<p>फ्लोचार्ट प्रतीकों में शब्द सामान्य कथन और समझने में आसान होने चाहिए। ये प्रोग्रामिंग भाषाओं से स्वतंत्र होने चाहिए।</p>	<p>फ्लोचार्ट में नाम और चर का उपयोग करने में सुसंगत रहें।</p>	<p>यदि फ्लोचार्ट बड़ा और जटिल हो जाता है तो प्रवाह रेखाओं को काँस करने से बचने के लिए कनेक्टर प्रतीकों का उपयोग किया जाना चाहिए।</p>
<p>विभिन्न पृष्ठों पर फ्लोचार्ट के भागों को जोड़ने के लिए उचित लेबल वाले कनेक्टरों का उपयोग किया जाना चाहिए।</p>		<p>फ्लोचार्ट में स्टार्ट और स्टॉप पॉइंट होने चाहिए।</p>

चित्र 4.27 फ्लोचार्ट नियम और दिशानिर्देश

फ़्लोचार्ट का उदाहरण

संख्याओं के औसत की गणना के लिए एक एल्गोरिथम पर विचार करें। फ़्लोचार्ट को नीचे दिए गए चित्र में दर्शाया गया है। पूर्णांक चर  $n$  औसत की गणना के लिए माने गए मानों की संख्या को दर्शाता है। गिनती अभी तक एक और पूर्णांक चर है जो किसी भी समय संसाधित किए गए मानों की संख्या का प्रतिनिधित्व करता है। मानों को संग्रहीत करने के लिए, संख्या एक पूर्णांक चर है।



चित्र 4.28 10 पूर्णांक संख्याओं का औसत ज्ञात करने के लिए फ़्लोचार्ट

फ़्लोचार्ट के कुछ लाभ निम्नलिखित हैं:

#### संक्षेपण:

- फ्लोचार्ट का उपयोग नए प्रोग्राम और सॉफ्टवेयर सिस्टम को डिजाइन करने में कार्यशील मॉडल के रूप में किया जाता है।

#### दस्तावेजीकरण:

- कार्यक्रम के दस्तावेजीकरण में गतिविधियाँ शामिल होती हैं, जैसे किसी कार्यक्रम के सभी संबंधित रिकॉर्ड को एकत्रित करना, व्यवस्थित करना, संग्रहीत करना और बनाए रखना।

#### कोडिंग:

- फ्लोचार्ट उच्च-स्तरीय भाषा में वास्तविक कोड लिखने में प्रोग्रामर का मार्गदर्शन करता है, जो कि शीघ्रता से विकसित एक त्रुटि-मुक्त प्रोग्राम देने वाला है।

#### डिवर्गिंग:

- किसी प्रोग्राम में त्रुटियों का पता कंप्यूटर पर उसके निष्पादन के बाद ही चलता है।
- इन त्रुटियों को बग कहा जाता है और इन त्रुटियों को दूर करने की प्रक्रिया को डिवर्गिंग कहा जाता है।
- डिवर्गिंग प्रक्रिया में, फ्लोचार्ट प्रोग्राम से बग्स का पता लगाने, पता लगाने और हटाने में एक महत्वपूर्ण उपकरण के रूप में कार्य करता है।

#### संचार:

- एक फ्लोचार्ट एक कार्यक्रम का एक सचित्र प्रतिनिधित्व है। इसलिए, अन्य प्रोग्रामर्स/लोगों को प्रोग्राम के तर्क को समझाने के लिए यह एक उत्कृष्ट संचार तकनीक है।

#### विक्षेपण:

- संबंधित फ्लोचार्ट की सहायता से किसी तार्किक समस्या का प्रभावी विक्षेपण आसानी से किया जा सकता है।

#### परिक्षण:

- एक फ्लोचार्ट एक प्रोग्रामर के हाथ में एक महत्वपूर्ण उपकरण है, जो उसे कार्यक्रमों के व्यवस्थित परीक्षण के लिए परीक्षण डेटा को डिजाइन करने में मदद करता है।

चित्र 4.29 फ्लोचार्ट के कुछ लाभ

फ्लोचार्ट के कुछ नुकसान निम्नलिखित हैं:

फ्लोचार्ट के विभिन्न नुकसान इस प्रकार हैं:

फ्लोचार्ट बनाना एक समय लेने वाली और श्रमसाध्य प्रक्रिया है, विशेष रूप से बड़ी, जटिल समस्याओं के लिए।

फ्लोचार्ट पुनर्लेखन और भी कठिन और समय लेने वाला है। मौजूदा फ्लोचार्ट में किसी भी नए चरण को शामिल करना बेहद मुश्किल है, और फ्लोचार्ट को फिर से बनाना ही एकमात्र उपाय है।

कोई मानक दिशानिर्देश नहीं हैं जो किसी भी फ्लोचार्ट में शामिल किए जाने वाले विवरण के स्तर को निर्दिष्ट करते हैं।

जब किसी एल्गोरिथ्म में जटिल शाखाएं और लूप होते हैं तो फ्लोचार्ट बनाना बेहद मुश्किल हो जाता है।

#### घटना सूचियाँ

ईवेंट सूचियाँ बाहरी घटनाओं की सूची होती हैं जो बाहरी दुनिया में होती हैं और सिस्टम पर प्रभाव डालती हैं, जैसे कि कुछ टर्मिनेटर द्वारा उत्पन्न ईवेंट। संदर्भ आरेख और घटना सूचियों को किसी भी क्रम में तब तक सूचीबद्ध किया जा सकता है जब तक कि दोनों इकाइयाँ निर्मित हों और एक दूसरे के अनुरूप हों।

निम्नलिखित घटनाओं के उदाहरण हैं:

प्रवाह - ये घटनाएँ तब होती हैं जब बाहरी दुनिया और सिस्टम के बीच कुछ डेटा प्रवाहित होता है। उदाहरण के लिए, कोई ग्राहक आरक्षण करता है।

बी) अस्थायी - अस्थायी घटनाएँ तब होती हैं जब कोई घटना किसी समय के परिणामस्वरूप होती है। उदाहरण के लिए, प्रबंधन एक साप्ताहिक रिपोर्ट का अनुरोध करता है।

ग) नियंत्रण - नियंत्रण घटनाएँ एक अस्थायी घटना का एक विशेष मामला है, एक बाहरी उत्तेजना जो किसी अप्रत्याशित समय पर होती है। उदाहरण के लिए, क्रेडिट कार्ड का सत्यापन।

घटनाओं को हमेशा टर्मिनेटर की नजर से देखा जाता है। घटनाएँ सिस्टम के बाहर होती हैं, और समय बीतने के कारण कोई घटना घटित हो सकती है।

## सारांश

- फ़्लोचार्ट का आरेखण एक बहुत समय लेने वाली प्रक्रिया है और विशेष रूप से बड़ी, जटिल समस्याओं के लिए श्रमसाध्य है।
- नियंत्रण घटनाएँ एक अस्थायी घटना का एक विशेष मामला है, एक बाहरी उत्तेजना जो समय में कुछ अप्रत्याशित बिंदु पर होती है।
- डिबगिंग। किसी प्रोग्राम में त्रुटियों का पता कंप्यूटर पर उसके निष्पादन के बाद ही चलता है। इन त्रुटियों को बग कहा जाता है और इन त्रुटियों को दूर करने की प्रक्रिया को डिबगिंग कहा जाता है। डिबगिंग प्रक्रिया में, फ़्लोचार्ट प्रोग्राम से बग्स का पता लगाने, पता लगाने और हटाने में एक महत्वपूर्ण उपकरण के रूप में कार्य करता है।
- फ्लो लाइन्स। प्रवाह रेखाएँ, जिनमें तीर के निशान होते हैं, का उपयोग फ़्लोचार्ट में प्रोग्राम लॉजिक के प्रवाह को इंगित करने के लिए किया जाता है। इन तीरों का उपयोग नियंत्रण के प्रवाह की दिशा को इंगित करने के लिए किया जाता है। इसका मतलब है कि ये बयान अगले बयान को निष्पादित करने का संकेत देते हैं।
- डेटा तत्वों में डेटा और नियंत्रण आइटम, आंतरिक या बाहरी डेटा स्टोर आदि के नाम और विवरण शामिल हैं।
- डेटा डिक्शनरी डेटा के बारे में जानकारी का केंद्रीकृत संग्रह है। यह डेटा के अर्थ और उत्पत्ति, अन्य डेटा के साथ इसके संबंध और उपयोग के लिए एक प्रारूप को संग्रहीत करता है। उपयोगकर्ता और सॉफ्टवेयर डिजाइनरों की सुविधा के लिए डेटा डिक्शनरी में सभी नामों की कठोर परिभाषाएं हैं।
- सॉफ्टवेयर प्रोग्राम का DFD (डेटा फ्लो डायग्राम) मॉडल और जब भी DFD को बदला या अपडेट किया जाता है, तो इसके अपडेट होने की उम्मीद है।
- सत्यापन योग्य होने के लिए, अमूर्तता के एक स्तर पर आवश्यकता विनिर्देशों को अमूर्तता के दूसरे स्तर पर संगत होना चाहिए। इन विशेषताओं में से अधिकांश, यदि सभी नहीं हैं, तो व्यक्तिपरक हैं और आवश्यकताओं के विनिर्देश की गुणवत्ता के निर्णायक मूल्यांकन के लिए आवश्यकताओं द्वारा संबोधित डोमेन में तकनीकी और परिचालन विशेषज्ञों द्वारा समीक्षा और विश्लेषण की आवश्यकता होती है।

## अभ्यास



नीचे दिए गए प्रश्नों के उत्तर दीजिये

फ्लोचार्ट बनाने के नियम क्या हैं?

स्यूडो संहिता के फायदे और नुकसान क्या हैं?

एसआरएस के महत्व की व्याख्या करें।

इस पर एक संक्षिप्त नोट लिखें:

एसआरएस के लिए आईईईई मानक

हिपो आरेख

संरचनात्मक डिजाइन

निर्णय तालिका

रिक्त स्थानों की पूर्ति कीजिये

सॉफ्टवेयर प्रोग्राम का \_\_\_\_\_ मॉडल और जब भी \_\_\_\_\_ को बदला या अपडेट किया जाता है तो इसके अपडेट होने की उम्मीद है।

\_\_\_\_\_ एक दस्तावेज है जो बताता है कि सॉफ्टवेयर क्या करेगा और उसे कैसा प्रदर्शन करना चाहिए। यह सभी हितधारकों (व्यवसाय और उपयोगकर्ताओं) की जरूरतों को पूरा करने के लिए उत्पाद द्वारा आवश्यक कार्यक्षमता का भी वर्णन करता है।

\_\_\_\_\_ ग्राहक और ठेकेदारों के बीच हुए समझौते के आधार पर बनाया गया है। इसमें उपयोग के मामले शामिल हो सकते हैं कि उपयोगकर्ता सॉफ्टवेयर सिस्टम के साथ कैसे इंटरैक्ट करेगा।

\_\_\_\_\_ स्थिरता और/या महत्व के अनुसार आवश्यकताओं दस्तावेज के संगठन और संरचना में स्थापित किया गया है।

## यूनिट 4.2: कस्टम और रैपिड एप्लिकेशन सॉफ्टवेयर

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. कस्टम एप्लिकेशन डेवलपमेंट और रैपिड एप्लिकेशन डेवलपमेंट के बीच प्राथमिक अंतरों पर चर्चा करने में

### 4.2.1 कस्टम अनुप्रयोग विकास

विशिष्ट उपयोगकर्ताओं के लिए कस्टम सॉफ्टवेयर एप्लिकेशन को डिज़ाइन करने, बनाने और तैनात करने की प्रक्रिया को कस्टम एप्लिकेशन डेवलपमेंट के रूप में जाना जाता है। कई स्कूल, स्वास्थ्य सेवा प्रदाता और व्यवसाय, उदाहरण के लिए, छात्रों, रोगियों और कर्मचारियों के लिए विभिन्न उद्देश्यों के लिए उपयोग करने के लिए अपने स्वयं के कस्टम पोर्टल बनाते हैं।

कस्टम अनुप्रयोग विकास के छह लाभ:

कस्टम सॉफ्टवेयर विशेष रूप से कंपनी के लिए डिज़ाइन किया गया है, व्यावसायिक प्रक्रियाओं को बदलने की आवश्यकता को समाप्त करते हुए दो एक पूर्व-विकसित सॉफ्टवेयर उत्पाद को समायोजित करता है।

सॉफ्टवेयर विशिष्ट व्यावसायिक कार्यों के लिए कई अनुप्रयोगों का उपयोग करने की आवश्यकता को दूर करते हुए, मौजूदा व्यावसायिक सॉफ्टवेयर सिस्टम पारिस्थितिकी तंत्र में मूल रूप से एकीकृत करता है।

उपयोगकर्ता हार्डवेयर खरीद पर पैसे बचा सकते हैं क्योंकि समाधान आपके बुनियादी ढांचे को ध्यान में रखकर बनाया गया है और इसमें अप्रयुक्त सुविधाएं शामिल नहीं हैं जो हार्डवेयर आवश्यकताओं को बढ़ाती हैं।

उपयोगकर्ताओं के पास मूल विकास टीम तक पूरी पहुंच है, जिससे किसी भी मुद्दे को जल्दी और कुशलता से हल किया जा सकता है।

कस्टम बिल्ट सॉफ्टवेयर होने से बाहरी ट्रेक व्यवसाय करना कम कर देते हैं क्योंकि हैकर्स के लिए केवल एक कंपनी द्वारा उपयोग किए जाने वाले सिस्टम में संध लगाने की कोशिश करने के लिए बहुत कम प्रोत्साहन होता है और संभावित हमलों पर प्रकाश डालने के लिए कोई सार्वजनिक दस्तावेज प्रदान नहीं किया जाता है।

व्यवसायों के पास सॉफ्टवेयर के बौद्धिक संपदा अधिकार हैं और मूल सॉफ्टवेयर डेवलपर अब इसे बनाए रखने में सक्षम नहीं होने पर भी आवश्यकतानुसार इसका उपयोग और सुधार करना जारी रख सकते हैं।

चित्र 4.30 कस्टम अनुप्रयोग विकास के लाभ

ग्राहक सॉफ्टवेयर विकास पेशेवर- कस्टम सॉफ्टवेयर का सबसे महत्वपूर्ण लाभ यह है कि इसमें ऐसी विशेषताएं शामिल हैं जो वाणिज्यिक सॉफ्टवेयर नहीं करती हैं। एक बार लागू होने के बाद, कस्टम-डिज़ाइन किए गए समाधान द्वारा हल की गई समस्याएं लागत के लायक हो सकती हैं।

ग्राहक सॉफ्टवेयर विकास विपक्ष - यदि कस्टम सॉफ्टवेयर समाधान को लागू करना सस्ता और सरल होता तो हर कोई ऐसा करता। इसमें शामिल लागतों और जोखिमों के कारण एक कस्टम सॉफ्टवेयर समाधान विकसित करना कठिन है।



**आमतौर**

अनुसूचित सॉफ्टवेयर बनाने के लिए पर्याप्त वित्तीय निवेश की आवश्यकता होती है। कमीशनरिंग कंपनी बिजनेस प्रक्रिया से जुड़ी सभी लागतों के लिए जिम्मेदार है। कुछ अनुसूचित समाधानों की कीमत पांच अंक या उससे अधिक होती है।



**जोखिम**

विकास प्रक्रिया के दौरान आपके कस्टम उत्पाद में अतिरिक्त सुविधाओं और कार्यों की खोज करना असामान्य नहीं है। संशोधन और संशोधन में अनिश्चितताएं आमतौर पर विकसित समय लगता है। बार-बार होने वाले परिवर्तनों के परिणामस्वरूप प्रोजेक्ट स्कोप लॉस और सॉफ्टवेयर हो सकता है जो आपकी आवश्यकताओं को पर्याप्त रूप से पूरा नहीं करता है।

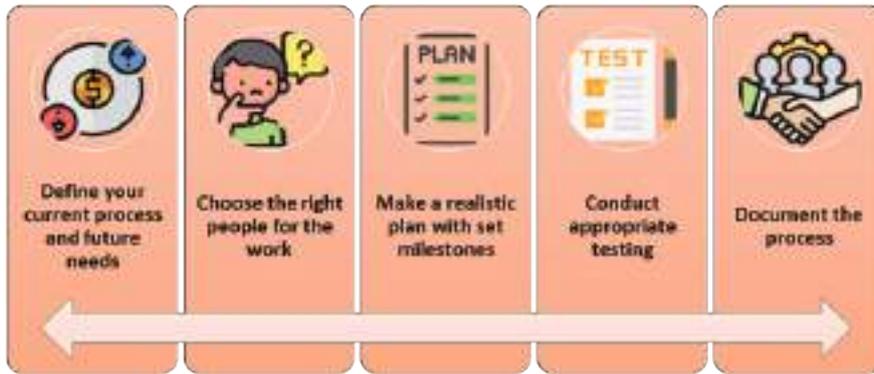
चित्र 4.31 ग्राहक सॉफ्टवेयर विकास विपक्ष

कस्टम सॉफ्टवेयर डेवलपमेंट कंपनी चुनने के लिए यहां कुछ संकेत दिए गए हैं:

अपने नेटवर्क और उद्योग के लोगों से रेफरल प्राप्त करें।	ऐसे डेवलपर्स की तलाश करें जो लीन, नॉन-बगगी कोड का उपयोग करते हैं।	पेश किए गए उद्योगों और परियोजना के आकार पर ध्यान केंद्रित करते हुए, उनके पिछले काम की जांच करें
समय पर डिलीवरी के इतिहास वाली कंपनियों पर ध्यान दें	सुनिश्चित करें कि दोतरफा संचार अच्छा है	बौद्धिक संपदा का मालिक कौन है?
सुनिश्चित करें कि वे एक इष्टतम उपयोगकर्ता अनुभव बना रहे हैं	सुरक्षा उपायों में सेंकना	विकास के बाद समर्थन अपेक्षाओं के बारे में स्पष्ट रहें

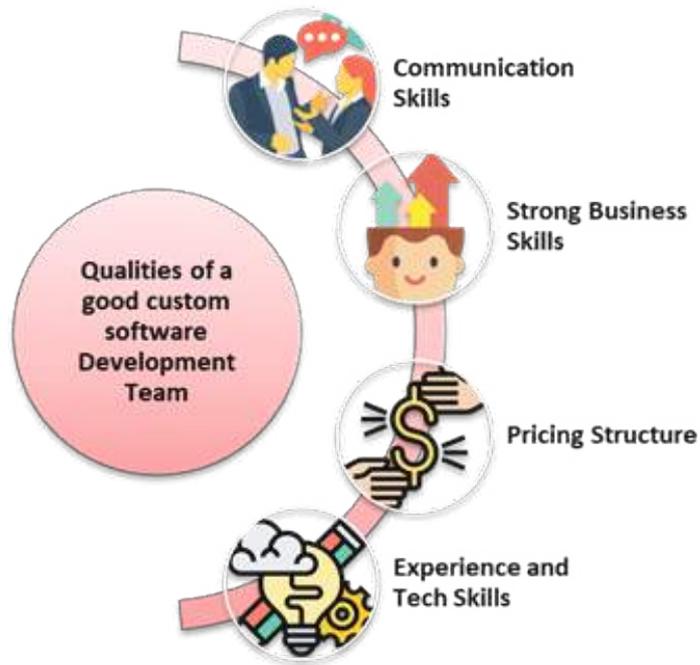
चित्र 4.32 कस्टम सॉफ्टवेयर डेवलपमेंट के चयन के लिए टिप्स

कस्टम सॉफ्टवेयर विकसित करने के लिए 5 सर्वोत्तम अभ्यास:



चित्र 4.33 कस्टम सॉफ्टवेयर विकसित करने के लिए अभ्यास

ग्राहक सॉफ्टवेयर डेवलपमेंट टीम गुण:



चित्र 4.34 कस्टम सॉफ्टवेयर डेवलपमेंट टीम के गुण

## 4.2.2 तीव्र अनुप्रयोग विकास

रैपिड एप्लिकेशन डेवलपमेंट (आरएडी) एक विकास मॉडल है जो लंबे विकास और परीक्षण चक्रों पर तेजी से प्रोटोटाइप और प्रतिक्रिया का समर्थन करता है। रैपिड एप्लिकेशन डेवलपमेंट डेवलपर्स को हर बार स्क्रीच से शुरू किए बिना सॉफ्टवेयर में कई पुनरावृत्तियों और अपडेट करने की अनुमति देता है। जब ऐसी प्रणाली बनाना आवश्यक हो जिसे 2-3 महीनों में मॉड्यूलर किया जा सके, तो आरएडी का उपयोग किया जाना चाहिए। इसका उपयोग तब किया जाना चाहिए जब मॉडलिंग के लिए डिजाइनरों की उच्च उपलब्धता हो और बजट उनकी सेवाओं की लागत के साथ-साथ स्वचालित कोड जनरेशन टूल की लागत को कवर करने के लिए पर्याप्त हो।

तीव्र अनुप्रयोग विकास के लाभ



चित्र 4.35 RAD के फायदे

निम्नलिखित आंकड़ा आरएडी के चार चरणों को प्रदर्शित करता है:



चित्र 4.36 RAD के चार चरण

#### चरण 1: बिजनेस मॉडलिंग

RAD मॉडल का व्यवसाय मॉडलिंग चरण कंपनी से व्यवसाय से संबंधित विभिन्न स्रोतों से जानकारी एकत्र करता है। इस जानकारी को तब एक उपयोगी विवरण में जोड़ा जाता है कि कैसे संसाधित होने पर डेटा का उपयोग किया जा सकता है, साथ ही साथ यह विशिष्ट जानकारी उद्योग के लिए क्या सफल बनाती है।

#### चरण 2: डेटा मॉडलिंग

डेटा मॉडलिंग चरण के दौरान, बिजनेस मॉडलिंग चरण के दौरान एकत्रित सभी सूचनाओं का विश्लेषण किया जाता है। जानकारी को विभिन्न समूहों में बांटा गया है जो विश्लेषण के परिणामस्वरूप कंपनी के लिए उपयोगी हो सकता है। प्रत्येक डेटा समूह की गुणवत्ता की सावधानीपूर्वक जांच की जाती है और उसका वर्णन किया जाता है। आरएडी मॉडल के इस चरण के दौरान, इन समूहों और बिजनेस मॉडलिंग चरण में परिभाषित उनकी उपयोगिता के बीच एक संबंध भी स्थापित किया जाता है।

#### चरण 3: प्रक्रिया मॉडलिंग

आरएडी मॉडल प्रक्रिया का प्रक्रिया मॉडलिंग चरण वह है जहां डेटा मॉडलिंग चरण के दौरान एकत्रित सभी सूचना समूहों को आवश्यक उपयोगी जानकारी में परिवर्तित किया जाता है। प्रक्रिया मॉडलिंग चरण के दौरान परिवर्तन और अनुकूलन किए जा सकते हैं, और डेटा सेट को और परिभाषित किया जा सकता है। इस चरण के दौरान, डेटा ऑब्जेक्ट को जोड़ने, हटाने या बदलने के लिए कोई विवरण भी बनाया जाता है।

#### चरण 4: अनुप्रयोग जनरेशन

एप्लिकेशन जनरेशन चरण में एकत्रित सभी सूचनाओं को कोड करना और सिस्टम का निर्माण करना शामिल है जिसका उपयोग प्रोटोटाइप बनाने के लिए किया जाएगा। अगले चरण में, डेटा मॉडल वास्तविक प्रोटोटाइप में तब्दील हो जाते हैं जिनका परीक्षण किया जा सकता है।

#### चरण 5: परीक्षण और कारोबार

परीक्षण और टर्नओवर चरण बनाए गए प्रोटोटाइप के समग्र परीक्षण में कम समय की अनुमति देता है। सबसे प्रभावी उत्पाद बनाने के लिए घटकों को जल्दी से पहचानने और अनुकूलित करने के लिए प्रत्येक मॉडल का अलग से परीक्षण किया जाता है। चूंकि अधिकांश तत्वों की पहले ही जांच की जा चुकी है, इसलिए आपके प्रोटोटाइप के साथ कोई बड़ी समस्या नहीं होनी चाहिए।

### 4.2.3 यूएमएल और ऑब्जेक्ट-ओरिएंटेड डिज़ाइन

#### वस्तु-उन्मुख डिज़ाइन:

हाल के वर्षों में, सॉफ्टवेयर विकास के लिए वस्तु-उन्मुख (OO) दृष्टिकोण लोकप्रियता में बढ़े हैं। अधिकांश नए विकास अब OO तकनीकों और भाषाओं के साथ किए गए हैं। OO सिस्टम कई फायदे प्रदान करते हैं। एक ओओ मॉडल समस्या डोमेन का बारीकी से प्रतिनिधित्व करता है, जिससे डिज़ाइन बनाना और समझना आसान हो जाता है। जैसे-जैसे आवश्यकताएं बदलती हैं, सिस्टम में वस्तुएं इन परिवर्तनों के प्रति कम प्रतिरोधी हो जाती हैं, जिससे अधिक आसानी से कार्यान्वित परिवर्तन संभव हो जाते हैं। इनहेरिटेन्स और ऑब्जेक्ट का डिज़ाइन में समस्या डोमेन संस्थाओं के साथ घनिष्ठ संबंध अधिक पुनः उपयोग को प्रोत्साहित करता है, अर्थात्, नए एप्लिकेशन मौजूदा मॉड्यूल का अधिक प्रभावी ढंग से उपयोग कर सकते हैं, विकास लागत और चक्र समय को कम कर सकते हैं। सोच और अमूर्तता के लिए समृद्ध संरचनाओं के साथ, वस्तु-उन्मुख दृष्टिकोण अधिक प्राकृतिक माना जाता है। सामान्य डिज़ाइन पैटर्न भी खोजे गए हैं, जिससे उच्च-स्तरीय पुनः प्रयोज्य की अनुमति मिलती है।

ऑब्जेक्ट-ओरिएंटेड डिज़ाइन अप्रोच मूल रूप से फंक्शन-ओरिएंटेड डिज़ाइन अप्रोच से भिन्न होता है, जिसका उपयोग अलग-अलग एबस्ट्रैक्शन के कारण होता है। यह सोचने और विभाजन करने के एक नए तरीके की आवश्यकता है। यह तर्क देना संभव है कि वस्तु-उन्मुख डिज़ाइनों के निर्माण के लिए वस्तु-उन्मुख शब्दों में सोचना आवश्यक है।

तब से, यह स्पष्ट हो गया है कि कार्य-उन्मुख विकास के पारंपरिक दृष्टिकोणों पर वस्तु-उन्मुख विकास के महत्वपूर्ण लाभ हैं। वस्तु-उन्मुख विकास के कुछ संभावित लाभ निम्नलिखित हैं:

**तेजी से विकास:**

• वस्तु-उन्मुख विकास को लंबे समय से तेजी से विकास के लिए अग्रणी माना जाता है। संभावित रूप से कम विकास समय के कई दावे सिद्धांत रूप में सही हैं यदि थोड़ा अधिक बताया गया है।

**पिछले कार्य का पुनः उपयोग:**

• यह साहित्य में सबसे अधिक उद्धृत लाभ है, विशेष रूप से व्यावसायिक पत्रिकाओं में। ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट सॉफ्टवेयर मॉड्यूल का उत्पादन करता है जिसे एक दूसरे में प्लग किया जा सकता है, जो नए प्रोग्राम के निर्माण की अनुमति देता है। हालांकि, ऐसा पुनः उपयोग आसानी से नहीं होता है। यह योजना और निवेश लेता है।

**बढ़ी हुई गुणवत्ता:**

• गुणवत्ता में वृद्धि मोटे तौर पर इस कार्यक्रम के पुनः उपयोग का उप-उत्पाद है। यदि किसी नए एप्लिकेशन के 90% में सिद्ध, मौजूदा घटक होते हैं, तो केवल शेष 10% कोड को खरोंच से परीक्षण करना होगा। उस अवलोकन का तात्पर्य दोषों में क्रम-परिमाण में कमी है।

**मॉड्यूलर वास्तुकला:**

• ऑब्जेक्ट-ओरिएंटेड सिस्टम में मॉड्यूलर डिजाइन के लिए एक प्राकृतिक संरचना होती है: ऑब्जेक्ट, सबसिस्टम, फ्रेमवर्क, और इसी तरह। इस प्रकार, वस्तु-उन्मुख विकास प्रणालियों को संशोधित करना आसान है। ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट सिस्टम को बिना किसी तोड़-फोड़ के मूलभूत तरीकों से बदला जा सकता है क्योंकि परिवर्तन बड़े करीने से होते हैं। हालांकि, ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट में कुछ भी गारंटी या आवश्यकता नहीं है कि उत्पादित कोड मॉड्यूलर होगा। ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट में एक मॉड्यूलर संरचना का निर्माण करने के लिए डिजाइन और कार्यान्वयन में समान स्तर की देखभाल की आवश्यकता होती है, क्योंकि यह किसी भी प्रकार के सॉफ्टवेयर विकास के लिए है।

**क्लाइंट/सर्वर अनुप्रयोग:**

• अपने स्वभाव से, क्लाइंट/सर्वर अनुप्रयोगों में नेटवर्क पर संदेशों का आगे और पीछे प्रसारण शामिल होता है, और ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट का ऑब्जेक्ट-मैसेज प्रतिमान क्लाइंट/सर्वर अनुप्रयोगों के भौतिक और वैचारिक वास्तुकला के साथ अच्छी तरह से मेल खाता है।

**समस्या डोमेन के लिए बेहतर मानचित्रण:**

• यह ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट के लिए एक स्पष्ट विजेता है, खासकर जब प्रोजेक्ट वास्तविक दुनिया में मैप करता है। चाहे वस्तुएं ग्राहकों, मशीनरी, बैंकों, सेंसर या कागज के टुकड़ों का प्रतिनिधित्व करती हों, वे एक स्वच्छ, आत्म-निहित निहितार्थ प्रदान कर सकती हैं जो स्वाभाविक रूप से मानव विचार प्रक्रियाओं में फिट बैठता है।

चित्र .4.37 वस्तु-उन्मुख विकास के लाभ

#### 4.2.4 यूएमएल (एकीकृत मॉडलिंग भाषा)

यूनिफाइड मॉडलिंग लैंग्वेज (यूएमएल) सॉफ्टवेयर सिस्टम कलाकृतियों को निर्दिष्ट करने, देखने, निर्माण करने और दस्तावेजीकरण के लिए एक मानक भाषा है। ऑब्जेक्ट मैनेजमेंट ग्रुप (ओएमजी) ने यूएमएल बनाया, और यूएमएल 1.0 विनिर्देश का एक मसौदा जनवरी 1994 में ओएमजी को प्रस्तावित किया गया था। ओएमजी एक सच्चे उद्योग मानक स्थापित करने के लिए लगातार काम कर रहा है।

यूएमएल यूनिफाइड मॉडलिंग लैंग्वेज का संक्षिप्त नाम है।

UML अन्य सामान्य प्रोग्रामिंग भाषाओं जैसे C++, Java, COBOL, इत्यादि से अलग है।

यूएमएल एक ग्राफिकल भाषा है जिसका उपयोग सॉफ्टवेयर ब्लूप्रिंट बनाने के लिए किया जाता है।

यूएमएल एक सामान्य-उद्देश्य वाली दृश्य मॉडलिंग भाषा है जिसका उपयोग सॉफ्टवेयर सिस्टम की कल्पना, निर्दिष्ट,

निर्माण और दस्तावेजीकरण के लिए किया जाता है।

हालांकि यूएमएल आमतौर पर सॉफ्टवेयर सिस्टम को मॉडल करने के लिए उपयोग किया जाता है, यह यहीं तक सीमित नहीं है। इसका उपयोग गैर-सॉफ्टवेयर सिस्टम का अनुकरण करने के लिए भी किया जाता है। उदाहरण के लिए, एक निर्माण इकाई में प्रक्रिया प्रवाह पर विचार करें।

यूएमएल एक प्रोग्रामिंग भाषा नहीं है, लेकिन यूएमएल आरेखों का उपयोग विभिन्न भाषाओं में कोड उत्पन्न करने के लिए किया जा सकता है। यूएमएल वस्तु-उन्मुख विश्लेषण और डिजाइन से निकटता से संबंधित है। कुछ मानकीकरण के बाद यूएमएल एक ओएमजी मानक बन गया है।

यूएमएल के लक्ष्य:

एक तस्वीर एक हजार शब्दों के बराबर होती है, और यह मुहावरा यूएमएल का पूरी तरह से वर्णन करता है। ऑब्जेक्ट-ओरिएंटेड अवधारणाएं लंबे समय तक यूएमएल की भविष्यवाणी करती हैं। वस्तु-उन्मुख विकास को व्यवस्थित और समेकित करने के लिए उस समय कोई मानक पद्धति नहीं थी।

यूएमएल ने उसी समय तस्वीर में प्रवेश किया। यूएमएल को विकसित करने के लिए कई लक्ष्य हैं, जिनमें से सबसे महत्वपूर्ण कुछ सामान्य-उद्देश्य वाली मॉडलिंग भाषा को परिभाषित करना है जिसका उपयोग सभी मॉडलर कर सकते हैं और यह समझने और उपयोग करने में भी आसान है।

यूएमएल आरेख न केवल डेवलपर्स के लिए, बल्कि व्यावसायिक उपयोगकर्ताओं, आम लोगों और किसी अन्य व्यक्ति के लिए भी बनाए जाते हैं जो सिस्टम को समझना चाहते हैं। इस प्रकार, यह स्पष्ट होना चाहिए कि यूएमएल एक विकास पद्धति नहीं है, बल्कि एक प्रक्रिया है जो एक सफल प्रणाली बनाने में मदद करती है। संक्षेप में, यूएमएल का लक्ष्य आज के जटिल वातावरण में सभी संभव व्यावहारिक प्रणालियों के मॉडलिंग के लिए एक सरल मॉडलिंग तंत्र प्रदान करना है।

यूएमएल बिल्डिंग ब्लॉक्स:

जैसा कि यूएमएल रीयल-टाइम सिस्टम का वर्णन करता है, एक वैचारिक मॉडल बनाना और फिर धीरे-धीरे आगे बढ़ना बहुत महत्वपूर्ण है। निम्नलिखित तीन प्रमुख तत्वों को सीखकर यूएमएल के वैचारिक मॉडल में महारत हासिल की जा सकती है-

यूएमएल बिल्डिंग ब्लॉक्स

बिल्डिंग ब्लॉक्स को जोड़ने के नियम

यूएमएल के सामान्य तंत्र

यूएमएल के निर्माण खंडों को इस प्रकार परिभाषित किया जा सकता है -



चित्र 4.38 UML की व्याख्या

चीजें यूएमएल के सबसे महत्वपूर्ण निर्माण खंड हैं। चीजें हो सकती हैं -

संरचनात्मक

व्यवहारिक

समूह बनाना

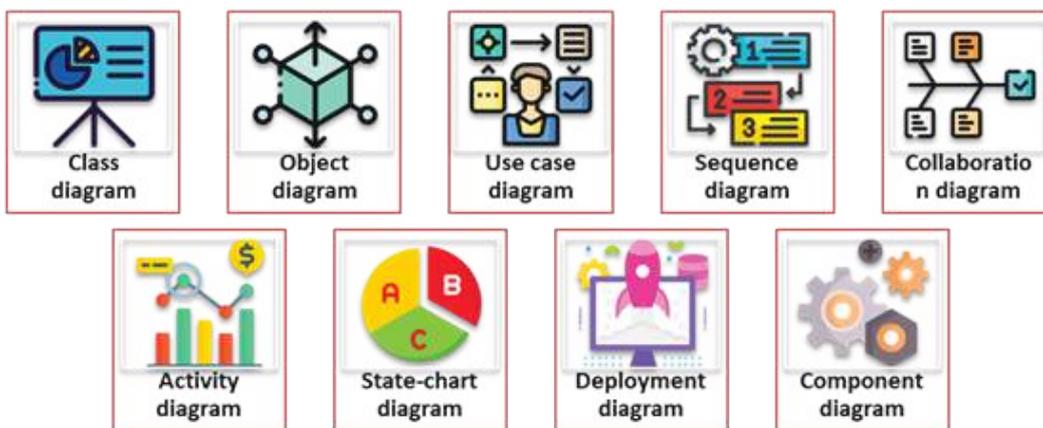
एनोटेशन

यूएमएल आरेख

यूएमएल आरेख संपूर्ण चर्चा का अंतिम परिणाम हैं। संपूर्ण यूएमएल आरेख बनाने के लिए सभी तत्वों और संबंधों का उपयोग किया जाता है और आरेख एक प्रणाली का प्रतिनिधित्व करता है।

यूएमएल आरेख का दृश्य प्रभाव पूरी प्रक्रिया का सबसे महत्वपूर्ण हिस्सा है। इसे पूरा करने के लिए अन्य सभी तत्वों का उपयोग किया जाता है।

यूएमएल में निम्नलिखित नौ आरेख शामिल हैं,



चित्र 4.39 UML चित्र

कक्षा आरेख

एक स्थिर आरेख वर्ग आरेख है। यह किसी एप्लिकेशन के स्थिर दृश्य का प्रतिनिधित्व करता है। क्लास डायग्राम का उपयोग न केवल सिस्टम के विभिन्न पहलुओं की कल्पना, वर्णन और दस्तावेजीकरण के लिए किया जाता है, बल्कि सॉफ्टवेयर एप्लिकेशन के लिए निष्पादन योग्य कोड बनाने के लिए भी किया जाता है। एक वर्ग आरेख एक वर्ग की विशेषताओं और संचालन के साथ-साथ सिस्टम की बाधाओं का वर्णन करता है। क्योंकि वे एकमात्र यूएमएल आरेख हैं जिन्हें सीधे ऑब्जेक्ट-ओरिएंटेड भाषाओं में मैप किया जा सकता है, ऑब्जेक्ट-ओरिएंटेड सिस्टम के मॉडलिंग में क्लास डायग्राम का व्यापक रूप से उपयोग किया जाता है। कक्षाओं, इंटरफेस, एसोसिएशन, सहयोग और बाधाओं का संग्रह कक्षा में दर्शाया गया है। आरेख।

एक संरचनात्मक आरेख इसका दूसरा नाम है।

कक्षा आरेखों का उद्देश्य

क्लास डायग्राम का उद्देश्य किसी एप्लिकेशन के स्टैटिक व्यू को मॉडल करना है। वर्ग आरेख एकमात्र ऐसे आरेख हैं जिन्हें सीधे वस्तु-उन्मुख भाषाओं के साथ मैप किया जा सकता है और इस प्रकार निर्माण के समय व्यापक रूप से उपयोग किया जाता है।

यूएमएल आरेख जैसे गतिविधि आरेख, और अनुक्रम आरेख केवल अनुप्रयोग का अनुक्रम प्रवाह दे सकते हैं; हालाँकि, वर्ग आरेख थोड़ा अलग है। यह कोडर समुदाय में सबसे लोकप्रिय यूएमएल आरेख है।

वर्ग आरेख के उद्देश्य को - . के रूप में संक्षेपित किया जा सकता है

एक आवेदन के स्थिर दृश्य का विश्लेषण और डिजाइन।

एक प्रणाली की जिम्मेदारियों का वर्णन करें।

घटक और परिनियोजन आरेखों के लिए आधार।

आगे और पीछे इंजीनियरिंग।

आम तौर पर, यूएमएल आरेख किसी ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषाओं के साथ सीधे मैप नहीं किए जाते हैं, लेकिन क्लास डायग्राम एक अपवाद है।

वर्ग आरेख स्पष्ट रूप से जावा, सी ++, आदि जैसी वस्तु-उन्मुख भाषाओं के साथ मानचित्रण दिखाता है। व्यावहारिक अनुभव से, एक वर्ग आरेख आमतौर पर निर्माण उद्देश्यों के लिए उपयोग किया जाता है।

संक्षेप में, यह कहा जा सकता है कि वर्ग आरेखों का उपयोग - . के लिए किया जाता है

प्रणाली के स्थिर दृष्टिकोण का वर्णन करना।

स्थिर दृश्य के तत्वों के बीच सहयोग दिखा रहा है।

सिस्टम द्वारा निष्पादित कार्यों का वर्णन करना।

वस्तु-उन्मुख भाषाओं का उपयोग करके सॉफ्टवेयर अनुप्रयोगों का निर्माण।

वस्तु आरेख

वस्तु आरेख वर्ग आरेखों से प्राप्त होते हैं, इसलिए वे वर्ग आरेखों पर निर्भर होते हैं। एक वर्ग आरेख को एक वस्तु आरेख द्वारा दर्शाया जाता है। मौलिक अवधारणाएं वर्ग और वस्तु आरेखों के समान हैं। ऑब्जेक्ट आरेख भी एक सिस्टम के स्थिर दृश्य का प्रतिनिधित्व करते हैं, लेकिन यह स्थिर दृश्य समय में एक विशिष्ट बिंदु पर सिस्टम का एक स्नैपशॉट है। ऑब्जेक्ट डायग्राम का उपयोग वस्तुओं के संग्रह और उनके संबंधों का एक उदाहरण बनाने के लिए किया जाता है।

वस्तु आरेखों का उद्देश्य

एक आरेख के उद्देश्य को व्यावहारिक रूप से लागू करने के लिए स्पष्ट रूप से समझा जाना चाहिए। वस्तु आरेखों के उद्देश्य वर्ग आरेखों के समान हैं।

अंतर यह है कि एक वर्ग आरेख कक्षाओं और उनके संबंधों से युक्त एक अमूर्त मॉडल का प्रतिनिधित्व करता है। हालाँकि, एक वस्तु आरेख एक विशेष क्षण में एक उदाहरण का प्रतिनिधित्व करता है, जो प्रकृति में ठोस है।

इसका मतलब है कि वस्तु आरेख वास्तविक सिस्टम व्यवहार के करीब है। इसका उद्देश्य किसी विशेष क्षण में सिस्टम के स्थिर दृश्य को कैप्चर करना है।

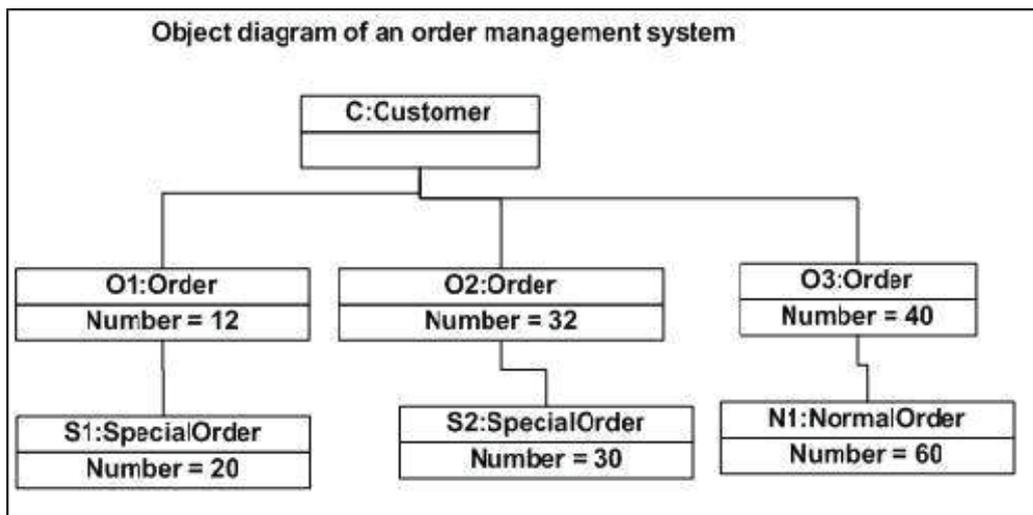
वस्तु आरेख के उद्देश्य को - . के रूप में संक्षेपित किया जा सकता है

आगे और पीछे इंजीनियरिंग।

एक प्रणाली के वस्तु संबंध

अंतःक्रिया का स्थिर दृष्टिकोण।

वस्तु व्यवहार और उनके संबंध को व्यावहारिक दृष्टिकोण से समझें



चित्र 4.40 ऑर्डर मैनेजमेंट सिस्टम का ऑब्जेक्ट आरेख

(Image Source: [https://www.tutorialspoint.com/uml/uml\\_object\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_object_diagram.htm))

#### घटक आरेख

घटक आरेख उनकी प्रकृति और व्यवहार में भिन्न होते हैं।

किसी सिस्टम की भौतिक विशेषताओं का प्रतिनिधित्व करने के लिए घटक आरेखों का उपयोग किया जाता है। अब प्रश्न यह है कि वास्तव में ये भौतिक पहलू क्या हैं? भौतिक पहलू वे तत्व हैं जो एक नोड में रहते हैं, जैसे निष्पादन योग्य, पुस्तकालय, फाइलें, दस्तावेज, और इसी तरह। घटक आरेखों का उपयोग सिस्टम घटकों के संगठन और संबंधों को दर्शाने के लिए किया जाता है।

इन आरेखों का उपयोग निष्पादन योग्य प्रणालियों के निर्माण में भी किया जाता है।

#### घटक आरेखों का उद्देश्य

यूएमएल में कंपोनेंट डायग्राम एक खास तरह का डायग्राम होता है। उद्देश्य अब तक चर्चा किए गए अन्य सभी आरेखों से भी अलग है। यह सिस्टम की कार्यक्षमता का वर्णन नहीं करता है, लेकिन यह उन कार्यात्मकताओं को बनाने के लिए उपयोग किए जाने वाले घटकों का वर्णन करता है।

इस प्रकार उस दृष्टिकोण से, एक सिस्टम में भौतिक घटकों की कल्पना करने के लिए घटक आरेखों का उपयोग किया जाता है। ये घटक पुस्तकालय, पैकेज, फाइलें आदि हैं।

घटक आरेखों को एक प्रणाली के स्थिर कार्यान्वयन दृश्य के रूप में भी वर्णित किया जा सकता है। स्थैतिक कार्यान्वयन एक विशेष क्षण में घटकों के संगठन का प्रतिनिधित्व करता है।

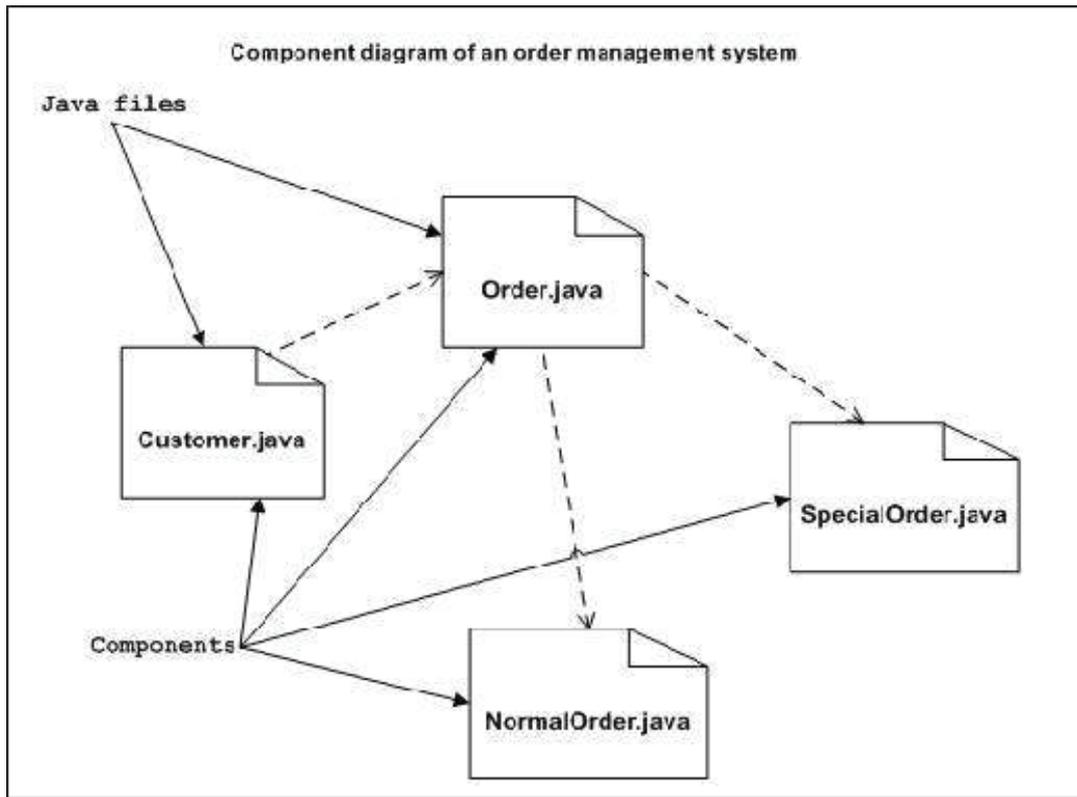
एक एकल घटक आरेख संपूर्ण प्रणाली का प्रतिनिधित्व नहीं कर सकता है लेकिन आरेखों के संग्रह का उपयोग संपूर्ण का प्रतिनिधित्व करने के लिए किया जाता है।

घटक आरेख के उद्देश्य को - . के रूप में संक्षेपित किया जा सकता है

सिस्टम के घटकों की कल्पना करें।

फॉरवर्ड और रिवर्स इंजीनियरिंग का उपयोग करके निष्पादन योग्य निर्माण करें।

घटकों के संगठन और संबंधों का वर्णन करें।



चित्र 4.41 एक आदेश प्रबंधन प्रणाली का घटक आरेख

(Image Source: [https://www.tutorialspoint.com/uml/uml\\_object\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_object_diagram.htm))

#### परिनियोजन आरेख

परिनियोजन आरेखों का उपयोग सिस्टम के भौतिक घटकों की टोपोलॉजी के साथ-साथ सॉफ्टवेयर घटकों के स्थानों को दर्शाने के लिए किया जाता है। सिस्टम के स्थिर परिनियोजन दृश्य का वर्णन करने के लिए परिनियोजन आरेखों का उपयोग किया जाता है।

नोट्स और उनके संबंधों को परिनियोजन आरेखों में दर्शाया गया है।

#### परिनियोजन आरेखों का उद्देश्य

परिनियोजन शब्द स्वयं आरेख के उद्देश्य का वर्णन करता है। परिनियोजन आरेख का उपयोग हार्डवेयर घटकों का वर्णन करने के लिए किया जाता है, जहां सॉफ्टवेयर घटक तैनात किए जाते हैं। घटक आरेख और परिनियोजन आरेख निकट से संबंधित हैं।

घटक आरेखों का उपयोग घटकों का वर्णन करने के लिए किया जाता है और परिनियोजन आरेख दिखाता है कि उन्हें हार्डवेयर में कैसे तैनात किया जाता है।

यूएमएल मुख्य रूप से एक सिस्टम के सॉफ्टवेयर कलाकृतियों पर ध्यान केंद्रित करने के लिए डिज़ाइन किया गया है। हालाँकि, ये दो आरेख विशेष आरेख हैं जिनका उपयोग सॉफ्टवेयर और हार्डवेयर घटकों पर ध्यान केंद्रित करने के लिए किया जाता है।

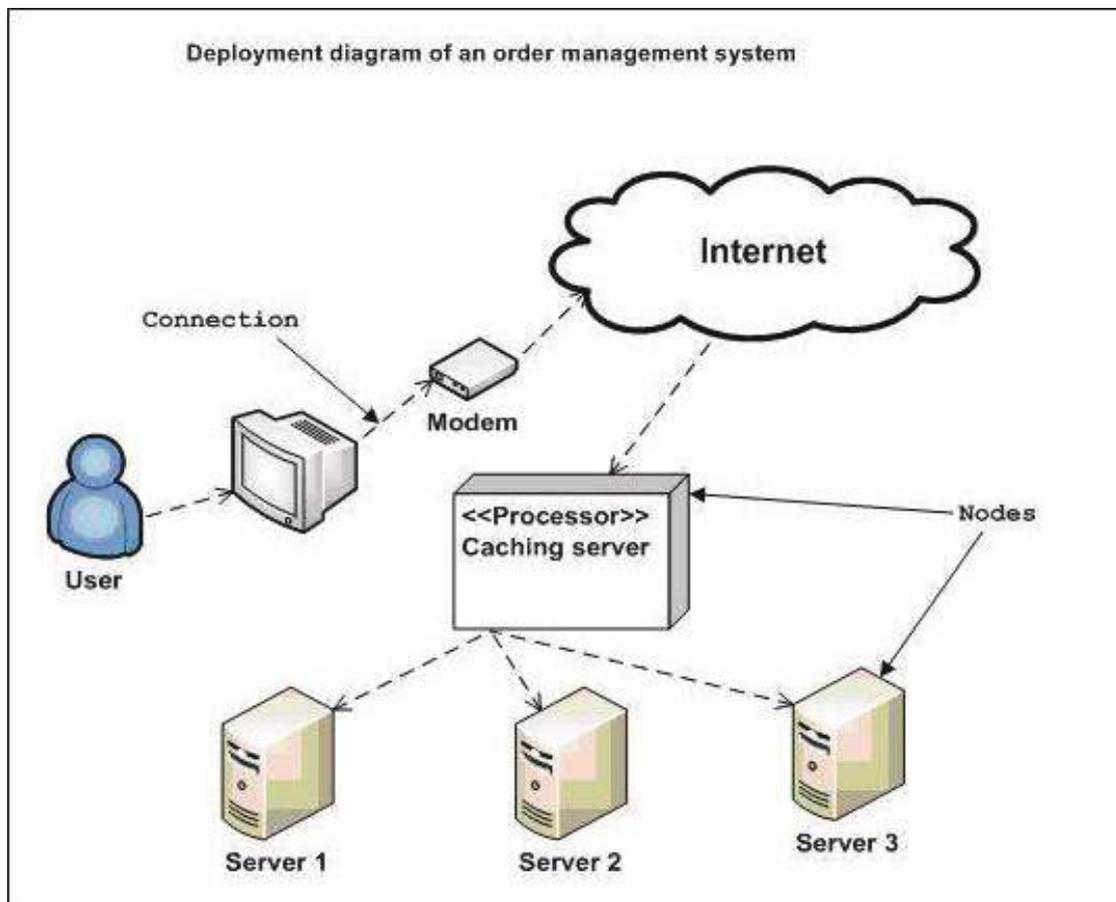
अधिकांश यूएमएल आरेख तार्किक घटकों को संभालने के लिए उपयोग किए जाते हैं लेकिन परिनियोजन आरेख एक सिस्टम के हार्डवेयर टोपोलॉजी पर ध्यान केंद्रित करने के लिए बनाए जाते हैं। सिस्टम इंजीनियरों द्वारा परिनियोजन आरेखों का उपयोग किया जाता है।

परिनियोजन आरेखों के उद्देश्य को - . के रूप में वर्णित किया जा सकता है

सिस्टम के हार्डवेयर टोपोलॉजी की कल्पना करें।

सॉफ्टवेयर घटकों को परिणियोजित करने के लिए प्रयुक्त हार्डवेयर घटकों का वर्णन करें।

रनटाइम प्रोसेसिंग नोइस का वर्णन करें।



चित्र 4.42 एक आदेश प्रबंधन प्रणाली का परिणियोजन आरेख

(Image Source: [https://www.tutorialspoint.com/uml/uml\\_deployment\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm))

केस डायग्राम का प्रयोग करें

एक प्रणाली को मॉडल करने के लिए, सबसे महत्वपूर्ण पहलू गतिशील व्यवहार को पकड़ना है। गतिशील व्यवहार का अर्थ है सिस्टम का व्यवहार जब वह चल रहा हो या संचालित हो रहा हो।

एक प्रणाली को मॉडल करने के लिए केवल स्थिर व्यवहार ही पर्याप्त नहीं है बल्कि स्थिर व्यवहार की तुलना में गतिशील व्यवहार अधिक महत्वपूर्ण है। यूएमएल में, गतिशील प्रकृति को मॉडल करने के लिए पांच आरेख उपलब्ध हैं और केस आरेख का उपयोग उनमें से एक है। अब जैसा कि हमें चर्चा करनी है कि उपयोग केस आरेख प्रकृति में गतिशील है, अंतःक्रिया करने के लिए कुछ आंतरिक या बाहरी कारक होने चाहिए।

इन आंतरिक और बाहरी एजेंटों को अभिनेता के रूप में जाना जाता है। एक उपयोग केस आरेख में अभिनेता, उपयोग के मामले और उनके संबंध होते हैं। आरेख का उपयोग किसी एप्लिकेशन के सिस्टम/सबसिस्टम को मॉडल करने के लिए किया जाता है। सिंगल यूज केस डायग्राम सिस्टम की एक विशेष कार्यक्षमता को कैच करता है।

इसलिए पूरे सिस्टम को मॉडल करने के लिए, कई उपयोग केस आरेखों का उपयोग किया जाता है।

उपयोग केस आरेख का उद्देश्य

यूज केस डायग्राम का उद्देश्य सिस्टम के डायनेमिक पहलू को कैप्चर करना है। हालाँकि, यह परिभाषा उद्देश्य का वर्णन करने के लिए बहुत सामान्य है, क्योंकि अन्य चार आरेखों (गतिविधि, अनुक्रम, सहयोग और स्टेटचार्ट) का भी एक ही उद्देश्य है। हम कुछ विशिष्ट उद्देश्य पर गौर करेंगे, जो इसे अन्य चार आरेखों से अलग करेगा।

आंतरिक और बाहरी प्रभावों सहित सिस्टम की आवश्यकताओं को इकट्ठा करने के लिए उपयोग केस आरेखों का उपयोग किया जाता है। ये आवश्यकताएं ज्यादातर डिजाइन आवश्यकताएं हैं। इसलिए, जब किसी प्रणाली का विश्लेषण उसकी कार्यप्रणाली को इकट्ठा करने के लिए किया जाता है, तो उपयोग के मामले तैयार किए जाते हैं और अभिनेताओं की पहचान की जाती है।

जब प्रारंभिक कार्य पूरा हो जाता है, तो बाहरी दृश्य प्रस्तुत करने के लिए उपयोग केस आरेखों को मॉडल किया जाता है।

संक्षेप में यूज केस डायग्राम के उद्देश्य इस प्रकार कहे जा सकते हैं -

एक। एक प्रणाली की आवश्यकताओं को इकट्ठा करने के लिए उपयोग किया जाता है।

बी। सिस्टम का बाहरी दृश्य प्राप्त करने के लिए उपयोग किया जाता है।

सी। प्रणाली को प्रभावित करने वाले बाहरी और आंतरिक कारकों की पहचान करें।

डी। अभिनेताओं की आवश्यकताओं के बीच बातचीत दिखाता है।

#### इंटरैक्शन आरेख

इंटरैक्शन शब्द से, यह स्पष्ट है कि आरेख का उपयोग मॉडल में विभिन्न तत्वों के बीच किसी प्रकार की बातचीत का वर्णन करने के लिए किया जाता है। यह अंतःक्रिया प्रणाली के गतिशील व्यवहार का एक भाग है।

इस संवादात्मक व्यवहार को यूएमएल में दो आरेखों द्वारा दर्शाया जाता है जिन्हें अनुक्रम आरेख और सहयोग आरेख के रूप में जाना जाता है। दोनों आरेखों के मूल उद्देश्य समान हैं।

अनुक्रम आरेख संदेशों के समय अनुक्रम पर जोर देता है और सहयोग आरेख उन वस्तुओं के संरचनात्मक संगठन पर जोर देता है जो संदेश भेजते और प्राप्त करते हैं।

#### इंटरैक्शन डायग्राम का उद्देश्य

इंटरैक्शन डायग्राम का उद्देश्य सिस्टम के इंटरैक्टिव व्यवहार की कल्पना करना है। बातचीत की कल्पना करना एक मुश्किल काम है। इसलिए, समाधान यह है कि बातचीत के विभिन्न पहलुओं को पकड़ने के लिए विभिन्न प्रकार के मॉडलों का उपयोग किया जाए।

अनुक्रम और सहयोग आरेखों का उपयोग गतिशील प्रकृति को पकड़ने के लिए किया जाता है लेकिन एक अलग कोण से।

अंतःक्रिया आरेख का उद्देश्य है -

एक। एक प्रणाली के गतिशील व्यवहार को पकड़ने के लिए।

बी। सिस्टम में संदेश प्रवाह का वर्णन करने के लिए।

सी। वस्तुओं के संरचनात्मक संगठन का वर्णन करने के लिए।

डी। वस्तुओं के बीच बातचीत का वर्णन करने के लिए।

#### अनुक्रम आरेख

एक अनुक्रम आरेख केवल अनुक्रमिक क्रम में वस्तुओं के बीच की बातचीत को दर्शाता है यानी जिस क्रम में ये बातचीत होती है। अनुक्रम आरेख को संदर्भित करने के लिए हम घटना आरेख या घटना परिदृश्य का भी उपयोग कर सकते हैं। अनुक्रम आरेख वर्णन करते हैं कि सिस्टम में ऑब्जेक्ट कैसे और किस क्रम में कार्य करते हैं। इन आरेखों का व्यापक रूप से व्यवसायियों और सॉफ्टवेयर डेवलपर्स द्वारा नए और मौजूदा सिस्टम के लिए आवश्यकताओं को दस्तावेज और समझने के लिए उपयोग किया जाता है।

## सहयोग आरेख

अनुक्रम आरेख के विपरीत, सहयोग आरेख वस्तुओं के बीच संबंधों को दर्शाता है। अनुक्रम आरेख और सहयोग आरेख समान जानकारी व्यक्त करते हैं लेकिन इसे अलग-अलग तरीकों से दिखाते हैं।

सहयोग आरेख के प्रारूप के कारण, वे विश्लेषण गतिविधियों के लिए बेहतर अनुकूल होते हैं (देखें गतिविधि: उपयोग-मामला विश्लेषण)। विशेष रूप से, वे कम संख्या में वस्तुओं की सरल बातचीत को चित्रित करने के लिए बेहतर अनुकूल होते हैं। हालाँकि, यदि वस्तुओं और संदेशों की संख्या बढ़ती है, तो आरेख को पढ़ना कठिन हो जाता है। इसके अलावा, अतिरिक्त वर्णनात्मक जानकारी जैसे समय, निर्णय बिंदु, या अन्य असंरचित जानकारी दिखाना मुश्किल है जिसे अनुक्रम आरेख में नोट्स में आसानी से जोड़ा जा सकता है। इसलिए, यहां कुछ उपयोग के मामले दिए गए हैं जिनके लिए हम एक सहयोग आरेख बनाना चाहते हैं:

वस्तुओं या भूमिकाओं के बीच मॉडल सहयोग जो उपयोग के मामलों और संचालन की कार्यक्षमता प्रदान करते हैं प्रणाली के स्थापत्य डिजाइन के भीतर मॉडल तंत्र

सहयोग के भीतर वस्तुओं और भूमिकाओं के बीच से गुजरने वाले संदेशों को दिखाने वाले इंटरैक्शन कैप्चर करें उपयोग के मामलों या संचालन के भीतर मॉडल वैकल्पिक परिदृश्य जिसमें विभिन्न वस्तुओं और इंटरैक्शन का सहयोग शामिल है

उपयोग के मामलों में भाग लेने वाली वस्तुओं (इसलिए कक्षाएं) की पहचान का समर्थन करें

सहयोग आरेख में प्रत्येक संदेश की एक क्रम संख्या होती है।

शीर्ष-स्तरीय संदेश क्रमांकित है 1. एक ही कॉल के दौरान भेजे गए संदेशों में एक ही दशमलव उपसर्ग होता है लेकिन 1, 2, आदि के प्रत्यय होते हैं, जब वे होते हैं।

सहयोग आरेख की सूचनाएं

वस्तुओं

एक वस्तु का प्रतिनिधित्व एक वस्तु प्रतीक द्वारा किया जाता है जो वस्तु का नाम और उसके वर्ग को रेखांकित करता है, एक कोलन द्वारा अलग किया जाता है:

ऑब्जेक्ट\_नाम: क्लास\_नाम

आप निम्नलिखित तरीकों से सहयोग आरेखों में वस्तुओं का उपयोग कर सकते हैं:

एक। सहयोग में प्रत्येक वस्तु का नाम दिया गया है और उसका वर्ग निर्दिष्ट है

बी। सभी वर्गों को प्रकट होने की आवश्यकता नहीं है

सी। एक वर्ग के एक से अधिक ऑब्जेक्ट हो सकते हैं

डी। किसी वस्तु का वर्ग अनिर्दिष्ट हो सकता है। आम तौर पर आप पहले वस्तुओं के साथ एक सहयोग आरेख बनाते हैं और बाद में उनकी कक्षाएं निर्दिष्ट करते हैं।

इ। वस्तुओं को अनामित किया जा सकता है, लेकिन यदि आप एक ही वर्ग की विभिन्न वस्तुओं के साथ भेदभाव करना चाहते हैं, तो आपको उनके विरुद्ध m का नाम देना चाहिए।

## स्टेट-चार्ट आरेख

आरेख का नाम ही आरेख के उद्देश्य और अन्य विवरणों को स्पष्ट करता है। यह एक प्रणाली में एक घटक के विभिन्न राज्यों का वर्णन करता है। स्टेट एक प्रणाली के निरंतर चार्ट के लिए विशिष्ट है।

एक स्टेट चार्ट आरेख एक स्टेट मशीन का वर्णन करता है। स्टेट मशीन को एक ऐसी मशीन के रूप में परिभाषित किया जा सकता है जो किसी वस्तु की विभिन्न अवस्थाओं को परिभाषित करती है और इन अवस्थाओं को गतिविधि या आंतरिक घटनाओं द्वारा नियंत्रित किया जाता है।

अगले यूनिट में समझाया गया गतिविधि आरेख, एक विशेष प्रकार का स्टेटचार्ट आरेख है। जैसा कि स्टेटचार्ट आरेख अवस्था को परिभाषित करता है, इसका उपयोग किसी वस्तु के जीवनकाल को मॉडल करने के लिए किया जाता है।

स्टेटचार्ट डायग्राम का उद्देश्य

एक स्टेटचार्ट आरेख पांच यूएमएल आरेखों में से एक है जो सिस्टम की गतिशील प्रकृति को मॉडल करने के लिए उपयोग किया जाता है। वे किसी वस्तु के जीवनकाल में उसकी विभिन्न अवस्थाओं को परिभाषित करते हैं और ये अवस्थाएँ घटनाओं द्वारा बदल जाती हैं। स्टेटचार्टग्राम प्रतिक्रियाशील प्रणालियों के मॉडल के लिए उपयोगी हैं। प्रतिक्रियाशील प्रणालियों को एक ऐसी प्रणाली के रूप में परिभाषित किया जा सकता है जो बाहरी या आंतरिक घटनाओं पर प्रतिक्रिया करती है।

स्टेटचार्ट आरेख एक स्टेट से दूसरे स्टेट में नियंत्रण के प्रवाह का वर्णन करता है। स्टेट को एक ऐसी स्थिति के रूप में परिभाषित किया जाता है जिसमें कोई वस्तु मौजूद होती है और जब कोई घटना शुरू होती है तो वह बदल जाती है। स्टेट चार्ट आरेख का सबसे महत्वपूर्ण उद्देश्य किसी वस्तु के निर्माण से लेकर समाप्ति तक के जीवनकाल को मॉडल करना है।

स्टेट चार्ट डायग्राम का उपयोग सिस्टम के फॉरवर्ड और रिवर्स इंजीनियरिंग के लिए भी किया जाता है। हालांकि, मुख्य उद्देश्य प्रतिक्रियाशील प्रणाली को मॉडल करना है।

स्टेट चार्ट आरेखों का उपयोग करने के मुख्य उद्देश्य निम्नलिखित हैं -

एक प्रणाली के गतिशील पहलू को मॉडल करने के लिए।

एक प्रतिक्रियाशील प्रणाली के जीवनकाल को मॉडल करने के लिए।

किसी वस्तु के जीवन काल में उसकी विभिन्न अवस्थाओं का वर्णन करना।

किसी वस्तु की अवस्थाओं को मॉडल करने के लिए एक राज्य मशीन को परिभाषित करें।

## सारांश



- विशिष्ट उपयोगकर्ताओं के लिए कस्टम सॉफ्टवेयर एप्लिकेशन को डिजाइन करने, बनाने और तैनात करने की प्रक्रिया को कस्टम एप्लिकेशन विकास के रूप में जाना जाता है।
- UML को ऑब्जेक्ट मैनेजमेंट ग्रुप (OMG) द्वारा बनाया गया था और UML 1.0 स्पेसिफिकेशन ड्राफ्ट जनवरी 1997 में OMG को प्रस्तावित किया गया था।
- गतिविधि आरेख का मूल उपयोग अन्य चार यूएमएल आरेखों के समान है। विशिष्ट उपयोग नियंत्रण प्रवाह को एक गतिविधि से दूसरे में मॉडल करना है। इस नियंत्रण प्रवाह में संदेश शामिल नहीं हैं।
- एक गतिविधि आरेख प्रणाली के गतिविधि प्रवाह को मॉडलिंग करने के लिए उपयुक्त है। एक एप्लिकेशन में कई सिस्टम हो सकते हैं।
- गतिविधि आरेख मुख्य रूप से एक फ्लोचार्ट के रूप में उपयोग किए जाते हैं जिसमें सिस्टम द्वारा की जाने वाली गतिविधियों से होता है। गतिविधि आरेख बिल्कुल फ्लोचार्ट नहीं हैं क्योंकि उनके पास कुछ अतिरिक्त क्षमताएं हैं।
- गतिविधि आरेखों का मूल उद्देश्य अन्य चार आरेखों के समान है। यह सिस्टम के गतिशील व्यवहार को कैप्चर करता है। अन्य चार आरेखों का उपयोग संदेश प्रवाह को एक ऑब्जेक्ट से दूसरे में दिखाने के लिए किया जाता है, लेकिन एक गतिविधि आरेख का उपयोग संदेश प्रवाह को एक गतिविधि से दूसरे में दिखाने के लिए किया जाता है।
- स्टेट चार्ट आरेखों का उपयोग स्टेट और सिस्टम पर काम करने वाली घटनाओं को मॉडल करने के लिए किया जाता है। किसी प्रणाली को लागू करते समय, किसी वस्तु के विभिन्न स्टेट को स्पष्ट करना बहुत महत्वपूर्ण है, इसके जीवनकाल के दौरान और स्टेट चार्ट आरेखों का उपयोग इस उद्देश्य के लिए किया जाता है।
- लिंक ऑब्जेक्ट्स और एक्टर्स को जोड़ते हैं और एसोसिएशन के उदाहरण हैं और प्रत्येक लिंक क्लास डायग्राम में एक एसोसिएशन से मेल खाता है।
- परिस्थितियों को मॉडल करने के लिए, हम यूएमएल में गार्ड का उपयोग करते हैं। उनका उपयोग तब किया जाता है जब हमें किसी शर्त को पूरा करने के बहाने संदेशों के प्रवाह को प्रतिबंधित करने की आवश्यकता होती है। सॉफ्टवेयर डेवलपर्स को सिस्टम या किसी विशेष प्रक्रिया से जुड़ी बाधाओं को जानने में गार्ड महत्वपूर्ण भूमिका निभाते हैं।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

यूएमएल द्वारा आप क्या समझते हैं?

UML के सबसे महत्वपूर्ण बिल्डिंग ब्लॉक क्या हैं?

ऑब्जेक्ट-ओरिएंटेड डेवलपमेंट के क्या फायदे हैं?

RAD में कितने चरण हैं?

रिक्त स्थान की पूर्ति कीजिये

\_\_\_\_\_ का उपयोग राज्यों और सिस्टम पर चलने वाली घटनाओं को मॉडल करने के लिए किया जाता है।

\_\_\_\_\_ का उपयोग मुख्य रूप से एक फ्लोचार्ट के रूप में किया जाता है जिसमें सिस्टम द्वारा की जाने वाली गतिविधियाँ शामिल होती हैं।

एक \_\_\_\_\_ आरेख केवल अनुक्रमिक क्रम में वस्तुओं के बीच अंतःक्रिया को दर्शाता है अर्थात् जिस क्रम में ये अंतःक्रियाएं होती हैं

एक \_\_\_\_\_ मॉडल समस्या क्षेत्र का बारीकी से प्रतिनिधित्व करता है, जिससे डिजाइन तैयार करना और समझना आसान हो जाता है।

MCQ प्रश्न:

यूएमएल इंटरफेस का उपयोग इसके लिए किया जाता है:

सभी वर्गों के लिए एपीआई

कक्षाओं में पुनः उपयोग करने के लिए निष्पादन योग्य तर्क

प्रोग्राम केवल जावा में

वस्तुओं के प्रकार के लिए आवश्यक सेवाएं निर्दिष्ट करें

निम्नलिखित में से किस यूएमएल आरेख में एक स्थिर दृश्य है?

गतिविधि

केस का प्रयोग करें

स्टेटचार्ट

सहयोग







IT - ITeS SSC  
NASSCOM

## 5. पुनः प्रयोज्य कोड डेवलपमेंट की अवधारणा

यूनिट 5.1 सॉफ्टवेयर परीक्षण



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. मैनुअल और स्वचालित परीक्षण प्रक्रिया के बारे में चर्चा करने में
2. आवश्यकताओं को पूरा करने के लिए विनिर्देशों की रूपांतरण प्रक्रिया को कोड में जांचने में

## यूनिट 5.1: सॉफ्टवेयर परीक्षण

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. सॉफ्टवेयर परीक्षण के अंतर्गत आने वाले सत्यापन और सत्यापन घटकों की पहचान करने में
2. एक परीक्षण योजना के घटकों की सूची बनाने में

### 5.1.1 आंतरिक या बाह्य उपयोगकर्ता सॉफ्टवेयर परीक्षण के लिए सत्यापन और मान्यकरण

#### सॉफ्टवेयर मान्यकरण

यह निर्धारित करने की प्रक्रिया कि क्या सॉफ्टवेयर उपयोगकर्ताओं की जरूरतों को पूरा करता है, सत्यापन के रूप में जाना जाता है। यह एसडीएलसी के बाद होता है। सत्यापन यह निर्धारित करता है कि सॉफ्टवेयर उन आवश्यकताओं को पूरा करता है या नहीं जिसके लिए इसे डिज़ाइन किया गया था

सत्यापन सुनिश्चित करता है कि विकास के तहत उत्पाद उपयोगकर्ताओं की जरूरतों को पूरा करेगा।

सत्यापन इस प्रश्न का उत्तर देता है, "क्या हम एक ऐसा समाधान तैयार कर रहे हैं जो उपयोगकर्ता की सभी सॉफ्टवेयर आवश्यकताओं को पूरा करने का प्रयास करता है?"

सत्यापन उपयोगकर्ता की आवश्यकताओं पर केंद्रित है।

विभिन्न प्रकार के सॉफ्टवेयर सत्यापन:

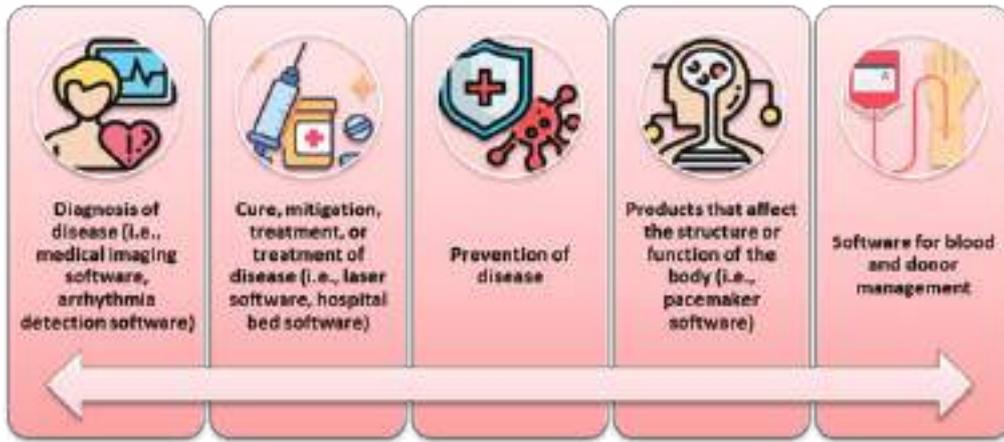


चित्र 5.1 विभिन्न प्रकार के सॉफ्टवेयर सत्यापन

#### चिकित्सा उपकरण सॉफ्टवेयर:

मेडिकल डिवाइस सॉफ्टवेयर में दोनों सॉफ्टवेयर शामिल होते हैं जिनका उपयोग किसी मेडिकल डिवाइस के घटक, भाग या एक्सेसरी के रूप में किया जाता है और सॉफ्टवेयर जो अपने आप में एक मेडिकल डिवाइस है।

इसमें उपयोग किए जाने वाले उत्पाद: चिकित्सा उपकरणों के उदाहरण हैं जिनमें ऐसे सॉफ्टवेयर होते हैं जिन्हें सत्यापन की आवश्यकता होती है।



चित्र 5.2 चिकित्सा उपकरण सॉफ्टवेयर

प्रोडक्शन सॉफ्टवेयर: एफडीए-विनियमित उत्पादों के निर्माण में उपयोग किए जाने वाले सॉफ्टवेयर और सिस्टम को प्रोडक्शन सॉफ्टवेयर कहा जाता है। इस समूह में शामिल हैं:

- सॉफ्टवेयर जो विनिर्माण उपकरण (यानी, पीएलसी, सीएनसी, निरीक्षण सॉफ्टवेयर) को नियंत्रित करता है
- सॉफ्टवेयर जो निर्माण प्रक्रिया का प्रबंधन करता है (यानी, कारक स्वचालन, उत्पादन निगरानी, सामग्री का बिल)
- प्रयोगशाला परीक्षण सॉफ्टवेयर
- लेबलिंग सॉफ्टवेयर

उत्पादन वातावरण के लिए प्रबंधन सॉफ्टवेयर (यानी, प्रबंधन प्रणाली का निर्माण)

विनिर्माण सॉफ्टवेयर जो महत्वपूर्ण संगणनाओं को स्वचालित करता है

ये संपूर्ण उदाहरण नहीं हैं; हालांकि, सत्यापन मुद्दों के लिए एफडीए चेतावनी पत्रों में इन श्रेणियों के उद्धरण शामिल हैं। गुणवत्ता प्रबंधन सॉफ्टवेयर: उत्पादों की गुणवत्ता सुनिश्चित करने के लिए सिस्टम में गुणवत्ता प्रबंधन सॉफ्टवेयर और सिस्टम का उपयोग किया जाता है।

FDA ने निम्नलिखित सभी उदाहरणों के लिए चेतावनी पत्र उद्धरण जारी किए हैं:

उत्पाद प्रबंधन सॉफ्टवेयर लौटाता है या याद करता है

शिकायत सॉफ्टवेयर

नियंत्रण सॉफ्टवेयर बदलें

निवारक रखरखाव सॉफ्टवेयर

इन्वेंटरी कंट्रोल सॉफ्टवेयर (यानी, ईआरपी)

दस्तावेज़ प्रबंधन सॉफ्टवेयर

विचलन ट्रैकिंग सॉफ्टवेयर

सीएपीए सॉफ्टवेयर

FDA-विनियमित रिकॉर्ड के लिए सॉफ्टवेयर: FDA-विनियमित रिकॉर्ड के लिए सॉफ्टवेयर को FDA-आवश्यक रिकॉर्ड बनाने, संशोधित करने, बनाए रखने, संग्रह करने, पुनर्प्राप्त करने या संचारित करने के लिए उपयोग किए जाने वाले सॉफ्टवेयर के रूप में परिभाषित किया गया है। एफडीए के अनुसार, इलेक्ट्रॉनिक रिकॉर्ड जमा किए जाने चाहिए।

FDA-विनियमित रिकॉर्ड सॉफ्टवेयर उदाहरणों में शामिल हैं:

आईआरबी रिकॉर्ड सॉफ्टवेयर

प्रतिकूल घटना रिपोर्टिंग सॉफ्टवेयर

अंग/ऊतक दाता रिकॉर्ड

कॉल सेंटर रिकॉर्ड सॉफ्टवेयर

सत्यापन रिकॉर्ड

प्रिस्क्रिप्शन ऑर्डर

पूर्ति सॉफ्टवेयर

क्लिनिकल परीक्षण रिकॉर्ड सॉफ्टवेयर

सॉफ्टवेयर सत्यापन: सॉफ्टवेयर सत्यापन किसी भी सॉफ्टवेयर की खामियों को खोजने के लिए समीक्षा करने की प्रक्रिया है। सत्यापन यह सुनिश्चित करने की प्रक्रिया है कि सॉफ्टवेयर बिना किसी दोष के अपने उद्देश्यों को पूरा करता है। यह निर्धारित करने की प्रक्रिया है कि विकसित उत्पाद सही है या नहीं। दस्तावेज़ की समीक्षा सॉफ्टवेयर की आवश्यकता और विश्लेषण चरण के रूप में शुरू हो सकती है, अंतिम उत्पाद SRS दस्तावेज़ होने के साथ। सॉफ्टवेयर सत्यापन का अभ्यास करने के लिए कई तरीके हैं, जैसे सहकर्मी समीक्षा, पूर्वाभ्यास, निरीक्षण, और इसी तरह, जो संभावित त्रुटियों की रोकथाम में हमारी सहायता कर सकते हैं, जिससे सॉफ्टवेयर विफलता हो सकती है।

सत्यापन सुनिश्चित करता है कि विकसित किया जा रहा उत्पाद डिजाइन विनिर्देशों के अनुसार है।

सत्यापन इस प्रश्न का उत्तर देता है- "क्या हम सभी डिजाइन विनिर्देशों का दृढ़ता से पालन करके इस उत्पाद को विकसित कर रहे हैं?"

सत्यापन डिजाइन और सिस्टम विनिर्देशों पर ध्यान केंद्रित करते हैं।

सत्यापन के तरीके:

पीयर रिव्यू - पीयर-रिव्यू विधि, खामियों की पहचान करने के लिए सत्यापन प्रक्रिया के दौरान दस्तावेजों या कार्यक्रमों/सॉफ्टवेयर की समीक्षा करने का सबसे सरल और सबसे अनौपचारिक तरीका है। इस पद्धति में, हम दूसरों को दस्तावेज़ या सॉफ्टवेयर प्रोग्राम वितरित करते हैं और उन्हें इस उम्मीद के साथ उनकी समीक्षा करने के लिए कहते हैं कि वे हमारे उत्पाद की गुणवत्ता पर प्रतिक्रिया प्रदान करेंगे और साथ ही प्रोग्राम/दस्तावेज़ में खामियों की पहचान करेंगे। SRS दस्तावेज़ सत्यापन, SDD सत्यापन और कार्यक्रम सत्यापन इस पद्धति के लिए सभी संभावित गतिविधियाँ हैं। इस पद्धति का उपयोग करके, समीक्षक अपनी टिप्पणियों या निष्कर्षों पर एक संक्षिप्त रिपोर्ट भी तैयार कर सकते हैं।

लाभ:

आप कोई महत्वपूर्ण संसाधन खर्च किए बिना कुछ अच्छे परिणामों की उम्मीद कर सकते हैं।

बी) यह बहुत ही कुशल और महत्वपूर्ण है।

नुकसान:

ए) यदि समीक्षक अनुभवहीन है, तो इससे खराब परिणाम हो सकते हैं

वॉक-थ्रू - जब पीयर-रिव्यू की तुलना में, वॉक-थ्रू सत्यापन के लिए एक अधिक औपचारिक और व्यवस्थित दृष्टिकोण है। वॉकथ्रू में, सॉफ्टवेयर दस्तावेज़ का लेखक दस्तावेज़ को 2 से 7 लोगों के समूह में प्रदर्शित करता है। तैयारी के मामले में प्रतिभागियों से कुछ भी नहीं मांगा जाता है। बैठक की तैयारी प्रस्तुतकर्ता की जिम्मेदारी है। सभी प्रतिभागियों को दस्तावेज़ (दस्तावेज़ों) की एक प्रति प्राप्त होती है। लेखक वॉक-थ्रू मीटिंग के दौरान सामग्री को इससे परिचित कराने के लिए प्रस्तुत करता है, और सभी उपस्थित लोगों को उनके कोई भी प्रश्न पूछने की अनुमति है।

लाभ:

यह संभावित दोषों की पहचान करने में हमारी सहायता कर सकता है।

इसका उपयोग दस्तावेजों पर सहयोग करने के लिए भी किया जा सकता है।

नुकसान:

लेखक दूसरों को छुपाते हुए अपने जुनून के कुछ विशिष्ट पहलुओं पर अनावश्यक रूप से जोर दे सकता है।

निरीक्षण - निरीक्षण सत्यापन का सबसे संरचित और औपचारिक तरीका है और इसे आम तौर पर निरीक्षण के रूप में जाना जाता है। निष्पक्ष मॉडरेटर के साथ तीन से छह लोगों का एक समूह बनाया जाता है। समूह में हर कोई इस तरह की समीक्षा करने के लिए खुले तौर पर, सक्रिय रूप से और दिशानिर्देशों के अनुसार संलग्न होता है। सभी को अपने दृष्टिकोण, संभावित कमियों और महत्वपूर्ण क्षेत्रों को साझा करने की अनुमति दी जा सकती है। बैठक के बाद, मॉडरेटर के आवश्यक सुझावों को अंतिम रिपोर्ट में शामिल किया जाता है।

लाभ:

एसआरएस, एसडीडी, आदि जैसे दस्तावेजों में संभावित दोषों या समस्याओं को खोजने के लिए यह बहुत प्रभावी हो सकता है।

महत्वपूर्ण निरीक्षण भी खामियों का पता लगाने और इन दस्तावेजों के सुधार में सहायता कर सकते हैं, जिससे सॉफ्टवेयर विकास जीवन चक्र में एक दोष के प्रसार को रोका जा सकता है।

नुकसान:

वे समय लेते हैं और अनुशासन की आवश्यकता होती है।

इसके लिए अधिक लागत की आवश्यकता होती है और कुशल परीक्षकों की भी आवश्यकता होती है।

सॉफ्टवेयर परीक्षण के क्षेत्र में, वाक्यांश सत्यापन और सत्यापन को कभी-कभी गलत समझा जाता है। आप इन शब्दों के विभिन्न उपयोगों और अर्थों से परिचित होंगे (या सामने आए हैं), और यह हमारा विनम्र प्रयास है कि हम उन्हें यहां यथासंभव स्पष्ट रूप से अलग करें।

मानदंड	सत्यापन	मान्यकरण
परिभाषा	एक विकास चरण से कार्य उत्पादों (अंतिम परिणाम के बजाय) की समीक्षा करने का अभ्यास यह देखने के लिए कि क्या वे चरण की आवश्यकताओं से मेल खाते हैं।	यह निर्धारित करने की प्रक्रिया कि क्या सॉफ्टवेयर विकास प्रक्रिया के दौरान या अंत में बताई गई व्यावसायिक आवश्यकताओं को पूरा करता है।
उद्देश्य	यह गारंटी देने के लिए कि उत्पाद विनिर्देशों और आवश्यकताओं के अनुसार बनाया गया है। इसे दूसरे तरीके से रखने के लिए, यह सुनिश्चित करने के लिए कि कार्य उत्पाद मानकों को पूरा करते हैं।	यह सुनिश्चित करने के लिए कि उत्पाद उपयोगकर्ताओं की आवश्यकताओं को पूरा करता है और यह कि विनिर्देश सही थे, शुरू करने के लिए। इसे दूसरे तरीके से रखने के लिए, यह दिखाने के लिए कि उत्पाद अपनी इच्छित सेटिंग में अपेक्षित प्रदर्शन करता है।
प्रश्न	क्या हम उत्पाद का निर्माण सही कर रहे हैं?	क्या हम सही उत्पाद बना रहे हैं?
मूल्यांकन आइटम	योजनाएँ, आवश्यकताएँ विशिष्टताएँ, डिज़ाइन विशिष्टताएँ, कोड, परीक्षण मामले	वास्तविक उत्पाद/सॉफ्टवेयर।

गतिविधि	समीक्षा प्रायोगिक प्रदर्शन निरीक्षण	टेस्टिंग
---------	---	----------

चित्र 5.1 सत्यापन और मान्यकरण

यह पूरी तरह से संभव है कि कोई उत्पाद सत्यापन पास करेगा फिर भी सत्यापन विफल हो जाएगा। यह तब हो सकता है, उदाहरण के लिए, एक उत्पाद विनिर्देशों के अनुसार बनाया गया है, फिर भी चश्मा उपयोगकर्ता की मांगों को पूरा नहीं करता है।

विश्वास करें लेकिन सत्यापित कर लें  
सत्यापित करें लेकिन मान्य भी करें

### 5.1.2 सॉफ्टवेयर परीक्षण

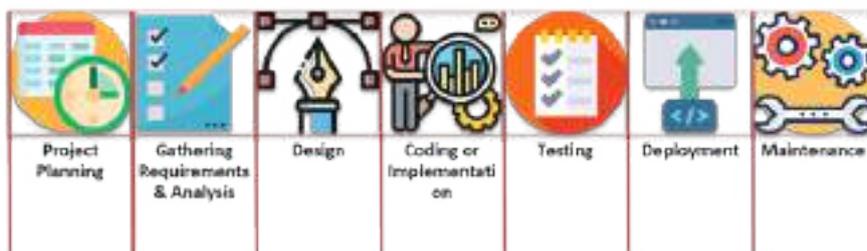
सॉफ्टवेयर परीक्षण उपयोगकर्ताओं और सिस्टम विनिर्देशों से एकत्रित आवश्यकताओं के विरुद्ध सॉफ्टवेयर का मूल्यांकन है। सॉफ्टवेयर विकास जीवन चक्र में चरण स्तर पर या प्रोग्राम कोड में माइड्रूल स्तर पर परीक्षण किया जाता है। सॉफ्टवेयर परीक्षण में विचार और सत्यापन शामिल हैं।

सॉफ्टवेयर परीक्षण प्रक्रिया



चित्र 5.4 सॉफ्टवेयर परीक्षण प्रक्रिया

सॉफ्टवेयर परीक्षण के 7 चरण



चित्र 5.5 सॉफ्टवेयर परीक्षण के चरण

### सॉफ्टवेयर परीक्षण का महत्व

पैसे बचाने में मदद करता है: सॉफ्टवेयर परीक्षण के कई लाभ हैं। व्यवसाय सॉफ्टवेयर परीक्षण सेवाओं की तलाश करने वाले प्राथमिक कारणों में से एक परियोजना लागत को कम करना है।

#### सुरक्षा:

यह एक और महत्वपूर्ण कारण है कि क्यों सॉफ्टवेयर परीक्षण से बचा जाना चाहिए।

बी) इसे सबसे कमजोर और संवेदनशील घटक माना जाता है। ऐसे कई उदाहरण हैं जिनमें उपयोगकर्ता की जानकारी और विवरण चोरी हो जाते हैं और व्यक्तिगत लाभ के लिए उपयोग किए जाते हैं।

यही कारण माना जाता है कि लोग अच्छी तरह से परीक्षित और भरोसेमंद उत्पादों की तलाश करते हैं।

जैसे ही किसी उत्पाद का परीक्षण किया जाता है, उपयोगकर्ता आश्वस्त हो सकता है कि उन्हें एक भरोसेमंद उत्पाद प्राप्त होगा। उपयोगकर्ता की व्यक्तिगत जानकारी सुरक्षित है। सॉफ्टवेयर परीक्षण की मदद से, उपयोगकर्ता ऐसे उत्पाद प्राप्त कर सकते हैं जो कमजोरियों से मुक्त हों।

3. उत्पाद की गुणवत्ता: विशिष्ट उत्पाद जीवन में आता है यह सुनिश्चित करने के लिए निम्नलिखित का पालन किया जाना चाहिए। उत्पाद की जरूरतों का पालन करना एक पूर्वापेक्षा है क्योंकि यह आवश्यक परिणाम प्राप्त करने में सहायता करता है। उत्पाद किसी न किसी रूप में उपयोगकर्ता के लिए उपयोगी होने चाहिए। इसे मूल्य के अपने वादे को पूरा करना चाहिए। नतीजतन, इसे ग्राहक अनुभव के लिए पूरी तरह से कार्य करना चाहिए। डिवाइस की संगतता की जांच करना भी आवश्यक है।

4. ग्राहक की संतुष्टि: उत्पाद के मालिक का प्राथमिक लक्ष्य ग्राहकों की संतुष्टि का उच्चतम स्तर प्रदान करना है। ग्राहक का विश्वास अर्जित करना कोई आसान काम नहीं है, खासकर अगर उत्पाद एक ही समय में काम कर रहा हो और गड़बड़ हो रहा हो। पहली छाप महत्वपूर्ण है, और यदि आप इसे प्रदान करने में विफल रहते हैं, तो उपयोगकर्ता किसी अन्य उत्पाद की तलाश करेंगे जो उनकी सभी आवश्यकताओं को पूरा करता हो।

5. विकास प्रक्रिया को बढ़ाना: गुणवत्ता आश्वासन त्रुटि के पुनरुत्पादन के लिए परिदृश्यों और त्रुटियों की एक विस्तृत श्रृंखला खोजने में आपकी सहायता कर सकता है। यह बहुत आसान है, और डेवलपर्स को इसे जल्द से जल्द ठीक करना चाहिए। इसके अलावा, सॉफ्टवेयर परीक्षकों को समानांतर में विकास टीम के साथ सहयोग करना चाहिए, जो विकास प्रक्रिया को तेज करने में मदद करता है।

6. नई सुविधाओं को जोड़ते समय आसान: कोड जितना पुराना और आपस में जुड़ा होता है, उसे बदलना उतना ही मुश्किल होता है। परीक्षण इस प्रवृत्ति को कम करने के लिए डेवलपर्स को आत्मविश्वास से नई सुविधाएँ जोड़ने की अनुमति देते हैं। एक नए डेवलपर के रूप में अपने कोडबेस के पुराने हिस्सों को बदलना भयानक हो सकता है, लेकिन परीक्षणों के साथ, आपको कम से कम पता चल जाएगा कि क्या आपने कुछ महत्वपूर्ण तोड़ दिया है। यह आपके सॉफ्टवेयर को भीड़ से अलग दिखने और प्रतियोगिता को मात देने में मदद करता है।

7. सॉफ्टवेयर के प्रदर्शन का निर्धारण: यदि उपयोगकर्ता ऐसे सॉफ्टवेयर या एप्लिकेशन खोजते हैं जिनका प्रदर्शन कम या कम है, तो यह उनकी बाज़ार प्रतिष्ठा को नुकसान पहुंचाएगा। यूजर्स किसी पर भरोसा नहीं करेंगे। आपके संगठन की प्रतिष्ठा को नुकसान होने की संभावना है। विशेषज्ञों के अनुसार, यह उतना महत्वपूर्ण नहीं है। हालाँकि, यदि उपयोगकर्ता पहले परीक्षण किए बिना सॉफ्टवेयर को बाज़ार में जारी करते हैं, और सॉफ्टवेयर का प्रदर्शन ग्राहकों की अपेक्षाओं या आवश्यकताओं से कम हो जाता है, तो उन्हें आश्वस्त करना मुश्किल होगा। इस प्रकार, सॉफ्टवेयर परीक्षण को एक सरल विकल्प माना जाता है क्योंकि यह सॉफ्टवेयर के प्रदर्शन को निर्धारित करने में सहायता करता है।

### 5.1.3 परीक्षण योजना के घटक

एक परीक्षण योजना बताती है कि परीक्षण कैसे किया जाएगा। यह एक दस्तावेज है जो सॉफ्टवेयर परीक्षण के लक्ष्य, दायरे और विधि को परिभाषित करता है। यह परीक्षण कार्यों और उन लोगों को निर्दिष्ट करता है जो उन्हें पूरा करेंगे, साथ ही परीक्षण की जाने वाली परीक्षण वस्तुओं और सुविधाओं को भी। यह परीक्षण वातावरण के साथ-साथ परीक्षण डिजाइन और माप तकनीकों का भी वर्णन करता है जिनका उपयोग किया जाएगा। यह ध्यान दिया जाना चाहिए कि एक अच्छी तरह से परिभाषित परीक्षण योजना परीक्षकों और उपयोगकर्ताओं के बीच एक समझौता है जो सॉफ्टवेयर में परीक्षण की भूमिका का वर्णन करता है। एक व्यापक परीक्षण योजना उन लोगों को सक्षम बनाती है जो परीक्षण समूह के सदस्य नहीं हैं, यह समझने के लिए कि उत्पाद सत्यापन की आवश्यकता क्यों है और इसे कैसे किया जाएगा। हालांकि, यदि परीक्षण योजना अधूरी है, तो यह परीक्षण करना असंभव हो सकता है कि विभिन्न ऑपरेटिंग सिस्टम पर स्थापित होने पर या अन्य सॉफ्टवेयर के साथ संयोजन में उपयोग किए जाने पर सॉफ्टवेयर कैसे काम करता है। आईईईई कुछ घटकों को निर्दिष्ट करता है जिन्हें इस समस्या से बचने के लिए परीक्षण योजना में शामिल किया जाना चाहिए। तालिका में इन घटकों की एक सूची है।

एक परीक्षण योजना के तालिका घटक

घटक	उद्देश्य
जिम्मेदारियाँ	विभिन्न लोगों को जिम्मेदारियाँ प्रदान करता है और उन्हें केंद्रित रखता है।
मान्यताओं	शेड्यूल की किसी भी गलत व्याख्या से बचा जाता है।
मान्यताओं	पूरी प्रक्रिया का सार प्रदान करता है और विशिष्ट परीक्षणों को रेखांकित करता है। परीक्षण की गुंजाइश, अनुसूची और अवधि भी उल्लिखित हैं।
संचार	एक संचार योजना (कौन, क्या, कब, कैसे, लोगों के बारे में) विकसित की जाती है।
संकट विश्लेषण	उन क्षेत्रों की पहचान करता है जो सफलता के लिए महत्वपूर्ण हैं
दोष रिपोर्टिंग	निर्दिष्ट करता है कि एक दोष को कैसे प्रलेखित किया जाना चाहिए ताकि यह फिर से तैयार हो सके और रिटेस्टेड और तय किया जा सके।
पर्यावरण	परीक्षण में उपयोग किए जाने वाले डेटा, इंटरफेस, कार्य क्षेत्र और तकनीकी वातावरण का वर्णन करता है। यह सब संभावित देरी के गलतफहमी और स्रोतों को कम करने या समाप्त करने के लिए निर्दिष्ट है।

सूची 5.2 एक परीक्षण योजना के घटक

एक सावधानीपूर्वक विकसित परीक्षण योजना प्रभावी परीक्षण निष्पादन, त्रुटियों का उचित विश्लेषण और त्रुटि रिपोर्ट तैयार करने की सुविधा प्रदान करती है। एक परीक्षण योजना विकसित करने के लिए, कई चरणों का पालन किया जाता है, जैसा कि नीचे सूचीबद्ध है।

परीक्षण योजना के उद्देश्य निर्धारित करें: किसी एक को विकसित करने से पहले एक परीक्षण योजना के उद्देश्य को समझना आवश्यक है। हालांकि, एक परीक्षण योजना के उद्देश्यों को निर्धारित करने से पहले, सॉफ्टवेयर उद्देश्यों को निर्धारित किया जाना चाहिए। ऐसा इसलिए है क्योंकि एक परीक्षण योजना के उद्देश्य सॉफ्टवेयर पर अत्यधिक निर्भर हैं। उदाहरण के लिए, यदि सॉफ्टवेयर का लक्ष्य सभी उपयोगकर्ता आवश्यकताओं को पूरा करना है, तो इस लक्ष्य को प्राप्त करने के लिए एक परीक्षण योजना बनाई जाती है।

एक परीक्षण मैट्रिक्स विकसित करें: एक परीक्षण मैट्रिक्स उन सॉफ्टवेयर घटकों की पहचान करता है जिनका परीक्षण किया जाएगा। यह उन परीक्षणों को भी निर्दिष्ट करता है जिन्हें यह सुनिश्चित करने के लिए किया जाना चाहिए कि ये घटक अच्छे कार्य क्रम में हैं। परीक्षण मैट्रिक्स का उपयोग परीक्षण प्रमाण के रूप में भी किया जाता है ताकि यह प्रदर्शित किया जा सके कि परीक्षण की आवश्यकता वाले सभी सॉफ्टवेयर घटकों के लिए एक परीक्षण मौजूद है। इसके अलावा, परीक्षण मैट्रिक्स का उपयोग परीक्षण पद्धति को इंगित करने के लिए किया जाता है जिसका उपयोग संपूर्ण सॉफ्टवेयर का परीक्षण करने के लिए किया जाएगा।

परीक्षण प्रशासनिक घटक विकसित करें: सॉफ्टवेयर परीक्षण जल्द से जल्द शुरू करने के लिए एक विशिष्ट समय सीमा के भीतर एक परीक्षण योजना तैयार की जानी चाहिए। एक परीक्षण योजना का प्रशासनिक घटक परीक्षण योजना को निष्पादित करने के लिए आवश्यक समय और संसाधनों (परीक्षण योजना के विकास में शामिल प्रशासनिक लोग) को निर्दिष्ट करता है। हालांकि, यदि सॉफ्टवेयर की कार्यान्वयन योजना (योजना जो बताती है कि सॉफ्टवेयर में प्रक्रियाओं को कैसे किया जाता है) बदलता है, तो परीक्षण योजना भी बदलती है। इस मामले में, परीक्षण योजना को पूरा करने का कार्यक्रम भी प्रभावित होता है।

परीक्षण योजना लिखें: एक परीक्षण योजना के घटक, जैसे कि इसके उद्देश्य, परीक्षण मैट्रिक्स और प्रशासनिक घटक, प्रलेखित हैं। इन सभी दस्तावेजों को तब एक व्यापक परीक्षण योजना बनाने के लिए संकलित किया जाता है। इन दस्तावेजों को दो श्रेणियों में बांटा गया है: अनौपचारिक और औपचारिक।



चित्र 5.6 एक परीक्षण में शामिल कदम

सभी दस्तावेजों को एकत्र किया जाता है और अनौपचारिक रूप से एक साथ रखा जाता है। सॉफ्टवेयर परीक्षण के लिए आवश्यक जानकारी निकालने के लिए सभी दस्तावेजों को परीक्षकों द्वारा पढ़ा गया था।

इसके विपरीत, दस्तावेजों से महत्वपूर्ण बिंदुओं को निकाला जाता है और सॉफ्टवेयर परीक्षण के दौरान आवश्यक महत्वपूर्ण जानकारी निकालने के लिए परीक्षकों के लिए एक साधारण औपचारिकता में एक साथ रखा जाता है। एक परीक्षण योजना को कई खंडों में विभाजित किया गया है, जो नीचे सूचीबद्ध हैं।

**अवलोकन:** उपयोग किए जाने वाले सॉफ्टवेयर के लक्ष्यों और कार्यों का वर्णन करता है। यह परीक्षण योजना के उद्देश्यों का भी वर्णन करता है, जैसे जिम्मेदारियों को परिभाषित करना, परीक्षण वातावरण की पहचान करना और परीक्षण योजना विकसित करने के लिए उपयोग किए जाने वाले स्रोतों की पूरी सूची प्रदान करना।

**परीक्षण का दायरा:** परीक्षण की जाने वाली सुविधाओं की विशेषताओं और संयोजनों को परिभाषित करता है। उपयोगकर्ता मैन्युअल और सिस्टम प्रलेखन ऐसी विशेषताओं के उदाहरण हैं। यह यह भी निर्दिष्ट करता है कि किन विशेषताओं और सुविधाओं के संयोजन का परीक्षण नहीं किया जाना है।

**परीक्षण के तरीके:** परीक्षण सुविधाओं और उनके संयोजनों के लिए आवश्यक परीक्षणों के प्रकारों को निर्दिष्ट करता है, जैसे प्रतिगमन परीक्षण और तनाव परीक्षण। यह परीक्षण डेटा के स्रोतों का भी वर्णन करता है और यह सुनिश्चित करने के लिए परीक्षण डेटा का उपयोग कैसे किया जा सकता है कि परीक्षण पर्याप्त है, जैसे सीमा या शून्य मानों का चयन। यह परीक्षा परिणामों को पहचानने और रिकॉर्ड करने की विधि का भी वर्णन करता है।

**परीक्षण चरण:** विभिन्न प्रकार के परीक्षणों की पहचान करता है, जैसे कि इकाई परीक्षण और एकीकरण परीक्षण, और इन परीक्षणों को करने के लिए उपयोग की जाने वाली प्रक्रिया का संक्षिप्त विवरण प्रदान करता है। इसके अलावा, यह उन परीक्षकों की पहचान करता है जो परीक्षण करेंगे और उपयोग किए जाने वाले डेटा स्रोत और प्रकार का विस्तृत विवरण प्रदान करते हैं। यह परीक्षण के परिणामों और इस चरण के दौरान शुरू या समाप्त किए गए कार्य उत्पादों के मूल्यांकन की प्रक्रिया का भी वर्णन करता है।

**परीक्षण वातावरण:** हार्डवेयर, सॉफ्टवेयर, स्वचालित परीक्षण उपकरण, ऑपरेटिंग सिस्टम, कंपाइलर और परीक्षण के लिए आवश्यक साइटों के साथ-साथ आवश्यक स्टाफिंग और प्रशिक्षण की पहचान करता है।

अनुसूची: परीक्षण गतिविधियों का विस्तार से वर्णन करता है और उपयुक्त व्यक्तियों को कार्य सौंपता है। यह परीक्षण गतिविधियों की निर्भरता के साथ-साथ उनके लिए समय सीमा को भी दर्शाता है।

अनुमोदन और वितरण: उन लोगों को सूचीबद्ध करता है जिन्होंने एक परीक्षण योजना और उसके निष्कर्षों को अपनी स्वीकृति दी है। यह उन लोगों को भी निर्दिष्ट करता है जिन्हें परीक्षण योजना दस्तावेज प्राप्त होंगे।

परीक्षण योजना पहचानकर्ता:

क्रमांक	पैरामीटर	विवरण
1	परीक्षण योजना पहचानकर्ता	अद्वितीय पहचान संदर्भ।
2	परिचय	परियोजना और दस्तावेज के बारे में एक संक्षिप्त परिचय।
3	आइटम का परीक्षण	एक परीक्षण आइटम एक सॉफ्टवेयर आइटम है जो परीक्षण के तहत एप्लिकेशन है।
4	परीक्षण किए जाने के लिए सुविधाएँ	एक फीचर जिसे टेस्ट वेयर पर टेस्ट करने की जरूरत है।
5	परीक्षण नहीं किया जाना चाहिए	परीक्षण के भाग के रूप में उन्हें शामिल न करने के कारणों और विशेषताओं की पहचान करें।
6	दृष्टिकोण	परीक्षण के लिए समग्र दृष्टिकोण के बारे में विवरण।
7	आइटम पास/विफल मानदंड	दस्तावेज किया गया है कि कोई सॉफ्टवेयर आइटम अपना परीक्षण पास कर चुका है या विफल हो गया है।
8	टेस्ट डिलिवरेबल्स	परीक्षण प्रक्रिया के हिस्से के रूप में वितरित किए जाने वाले डिलिवरेबल्स, जैसे कि परीक्षण योजना, परीक्षण विनिर्देशों और परीक्षण सारांश रिपोर्ट।
9	परीक्षण कार्य	परीक्षण की योजना और निष्पादित करने के लिए सभी कार्य।
10	पर्यावरणीय आवश्यकताएँ	हार्डवेयर, सॉफ्टवेयर, ओएस, नेटवर्क कॉन्फिगरेशन और आवश्यक उपकरण जैसी पर्यावरणीय आवश्यकताओं को परिभाषित करना।
11	जिम्मेदारियों	टीम के सदस्यों की भूमिकाओं और जिम्मेदारियों को सूचीबद्ध करता है।
12	स्टाफिंग और प्रशिक्षण की जरूरत है	वास्तविक स्टाफिंग आवश्यकताओं और किसी भी विशिष्ट कौशल और प्रशिक्षण आवश्यकताओं को कैप्चर करता है।
13	अनुसूची	महत्वपूर्ण परियोजना वितरण दिनांक और प्रमुख मील के पत्थर बताते हैं।
14	जोखिम और शमन	उच्च-स्तरीय परियोजना जोखिम और धारणाएं और प्रत्येक पहचाने गए जोखिम के लिए एक माइटिगेटिंग योजना।
15	स्वीकृति	दस्तावेज के सभी अनुमोदन, उनके शीर्षक और साइन-ऑफ तिथि को कैप्चर करता है।

सूची 5.3 परीक्षण योजना पहचानकर्ता

### 5.1.4 परीक्षण मामले

एक परीक्षण मामला एक दस्तावेज है जिसमें एक विशिष्ट आवश्यकता के अनुपालन को सत्यापित करने के लिए एक विशिष्ट परीक्षण परिदृश्य के लिए बनाए गए परीक्षण डेटा, पूर्व शर्त, अपेक्षित परिणाम और पोस्ट शर्तों का एक सेट होता है। इनपुट मूल्यों के एक सेट के आवेदन के बाद, प्रोग्राम का अंतिम परिणाम होता है और किसी बिंदु पर सिस्टम से बाहर निकलता है, जिसे निष्पादन पोस्ट स्थिति के रूप में भी जाना जाता है।

विशिष्ट परीक्षण केस पैरामीटर:



चित्र 5.7 परीक्षण केस पैरामीटर:

अच्छे टेस्ट केस लिखना:

परीक्षण मामलों को इस तरह से लिखें कि यदि संभव हो तो आप एक समय में केवल एक ही चीज़ का परीक्षण करें। परीक्षण के मामले अतिव्यापी या जटिल नहीं होने चाहिए। अपने परीक्षण मामलों को यथासंभव परमाणु बनाएं।

सुनिश्चित करें कि अच्छी और नकारात्मक दोनों घटनाओं को कवर किया गया है।

भाषा:

सरल और आसानी से समझ में आने वाली भाषा में लिखें।

निष्क्रिय आवाज के बजाय सक्रिय आवाज का प्रयोग करें: ऐसा करें, वह करें।

सटीक और सुसंगत नामों का प्रयोग करें (फॉर्मों, क्षेत्रों, आदि के)।

d. एक अच्छे परीक्षण मामले के लक्षण:

सटीक: सटीक उद्देश्य।

किफायती: कोई अनावश्यक कदम या शब्द नहीं।

ट्रेस करने योग्य: आवश्यकताओं का पता लगाने में सक्षम।

दोहराने योग्य: परीक्षण को बार-बार करने के लिए इस्तेमाल किया जा सकता है।

पुनः प्रयोज्य: यदि आवश्यक हो तो पुनः उपयोग किया जा सकता है।

### 5.1.5 परीक्षण दृष्टिकोण

टेस्ट दो दृष्टिकोणों के आधार पर आयोजित किए जा सकते हैं --



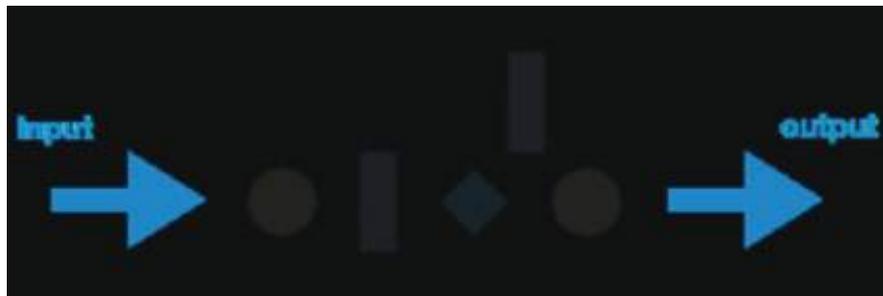
चित्र Fig 5.5 परीक्षण दृष्टिकोण

ब्लैक-बॉक्स परीक्षण तब होता है जब वास्तविक कार्यान्वयन को ध्यान में रखे बिना कार्यक्षमता का परीक्षण किया जाता है। दूसरी ओर, व्हाइट-बॉक्स परीक्षण में न केवल कार्यक्षमता का परीक्षण करना शामिल है, बल्कि यह विश्लेषण करना भी शामिल है कि इसे कैसे लागू किया जाता है।

संपूर्ण परीक्षण निर्दोष परीक्षण के लिए पसंदीदा तरीका है। इनपुट और आउटपुट मानों के बीच हर संभव मूल्य का परीक्षण किया जाता है। यदि मूल्यों की सीमा बड़ी है, तो वास्तविक दुनिया के परिदृश्य में प्रत्येक मूल्य का परीक्षण करना असंभव है।

ब्लैक-बॉक्स परीक्षण:

यह प्रोग्राम की कार्यक्षमता का परीक्षण करने के लिए किया जाता है। इसे 'व्यवहार' परीक्षण के रूप में भी जाना जाता है। इस मामले में, परीक्षक को इनपुट मानों और वांछित परिणामों का एक सेट दिया जाता है। यदि आउटपुट इनपुट प्रदान करने के बाद वांछित परिणामों से मेल खाता है, तो प्रोग्राम को 'ओके' माना जाता है, अन्यथा, इसे समस्याग्रस्त माना जाता है।



चित्र 5.9 ब्लैक बॉक्स परीक्षण

इस परीक्षण पद्धति में, कोड के डिजाइन और संरचना के बारे में परीक्षक को पता नहीं होता है, और परीक्षण इंजीनियर और अंतिम उपयोगकर्ता सॉफ्टवेयर पर इस परीक्षण का संचालन करते हैं।

**ब्लैक-बॉक्स परीक्षण तकनीक:**

**तुल्यता वर्ग** - इनपुट को समान समूहों में वर्गीकृत किया गया है। यदि किसी वर्ग का एक तत्व परीक्षा पास कर लेता है, तो पूरी कक्षा को उत्तीर्ण मान लिया जाता है।

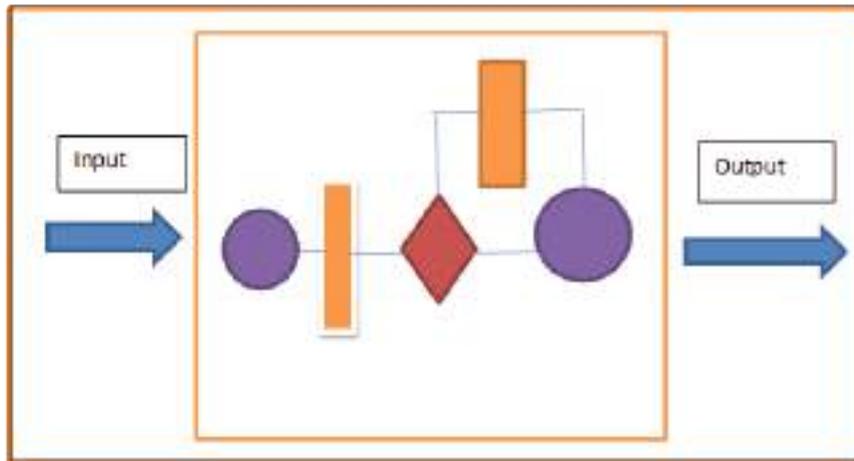
**सीमा मूल्य इनपुट** को उच्च और निम्न-अंत मूल्यों में विभाजित करते हैं। यदि ये मान परीक्षण पास करते हैं, तो यह माना जाता है कि उनके बीच के सभी मान भी पास हो जाएंगे।

**कारण-प्रभाव रेखांकन** - पूर्ववर्ती दोनों विधियों में, एक समय में केवल एक इनपुट मान का परीक्षण किया जाता है। कारण (इनपुट) - प्रभाव (आउटपुट) परीक्षण इनपुट मूल्यों के संयोजनों को व्यवस्थित रूप से परीक्षण करने की एक विधि है।

**जोड़ी-वार परीक्षण** - सॉफ्टवेयर का व्यवहार कई चरों से प्रभावित होता है। जोड़ीवार परीक्षण में उनके विभिन्न मूल्यों के लिए कई मापदंडों का परीक्षण किया जाता है।

**स्टेट-आधारित परीक्षण** में इनपुट के जवाब में सिस्टम की स्थिति को बदलना शामिल है। इन प्रणालियों के स्टेट और इनपुट का उपयोग उनका परीक्षण करने के लिए किया जाता है।

**व्हाइट-बॉक्स परीक्षण:** यह कोड दक्षता या संरचना में सुधार के लिए कार्यक्रम और इसके कार्यान्वयन का परीक्षण करने के लिए आयोजित किया जाता है। इसे 'संरचनात्मक' परीक्षण के रूप में भी जाना जाता है।



चित्र 5.10 व्हाइट-बॉक्स परीक्षण

इस परीक्षण पद्धति में, परीक्षक को कोड के डिजाइन और संरचना के बारे में पता होता है। कोड के प्रोग्रामर कोड पर इस परीक्षा का संचालन करते हैं।

**व्हाइट-बॉक्स परीक्षण तकनीक:**

**नियंत्रण-प्रवाह परीक्षण** - नियंत्रण-प्रवाह परीक्षण का उपयोग ऐसे परीक्षण मामले बनाने के लिए किया जाता है जो सभी कथनों और शाखा स्थितियों को कवर करते हैं। यह सुनिश्चित करने के लिए कि सभी कथनों को शामिल किया गया है, शाखा की शर्तों का परीक्षण सही और गलत दोनों के लिए किया जाता है।

**डेटा-प्रवाह परीक्षण** में शामिल सभी डेटा चर को कवर करने पर केंद्रित है। यह जांच करता है कि चर कहाँ घोषित और परिभाषित किए गए थे, साथ ही साथ उनका उपयोग या परिवर्तन कहाँ किया गया था।

### परीक्षण स्तर:

परीक्षण को विभिन्न एसडीएलसी स्तरों पर परिभाषित किया जा सकता है। सॉफ्टवेयर विकास के साथ-साथ परीक्षण होता है। अगले चरण पर जाने से पहले एक चरण का परीक्षण, सत्यापन और सत्यापन किया जाता है।

यह सुनिश्चित करने के लिए अलग परीक्षण किया जाता है कि सॉफ्टवेयर में कोई छिपी हुई बग या समस्या नहीं है। सॉफ्टवेयर का परीक्षण कई स्तरों पर किया जाता है

### इकाई मूल्यांकन

कोडिंग करते समय, प्रोग्रामर यह देखने के लिए प्रोग्राम की इकाई का परीक्षण करता है कि क्या यह त्रुटि रहित है। परीक्षण के लिए व्हाइट-बॉक्स परीक्षण दृष्टिकोण का उपयोग किया जाता है। यूनिट परीक्षण डेवलपर्स को यह सुनिश्चित करने की अनुमति देता है कि व्यक्तिगत प्रोग्राम इकाइयां ठीक से और त्रुटियों के बिना काम कर रही हैं।

### एकीकरण परीक्षण

भले ही सॉफ्टवेयर मॉड्यूल अपने आप ठीक से काम कर रहे हों, फिर भी यह निर्धारित करना आवश्यक है कि संयुक्त होने पर वे सही तरीके से काम करेंगे या नहीं। तर्क पारित करना और डेटा अद्यतन, आदि।

### सिस्टम परीक्षण

सॉफ्टवेयर को एक उत्पाद के रूप में संकलित किया जाता है और फिर इसका संपूर्ण परीक्षण किया जाता है। यह निम्नलिखित में से एक या अधिक परीक्षणों का उपयोग करके पूरा किया जा सकता है:

कार्यक्षमता परीक्षण सॉफ्टवेयर की सभी विशेषताओं की आवश्यकताओं के विरुद्ध तुलना करता है।

प्रदर्शन परीक्षण - यह परीक्षण सॉफ्टवेयर की दक्षता को स्थापित करता है। यह किसी कार्य को पूरा करने के लिए सॉफ्टवेयर की दक्षता और औसत समय का मूल्यांकन करता है। लोड टेस्टिंग और स्ट्रेस टेस्टिंग प्रदर्शन परीक्षण के तरीके हैं जिसमें प्रोग्राम को महत्वपूर्ण उपयोगकर्ता और विभिन्न वातावरणों में डेटा लोड के अधीन किया जाता है।

सुरक्षा और सुवाह्यता - ये परीक्षण तब किए जाते हैं जब सॉफ्टवेयर विभिन्न प्लेटफार्मों पर काम करने के लिए होता है और कई व्यक्तियों द्वारा एक्सेस किया जाता है।

स्वीकृति परीक्षण एक प्रकार का परीक्षण है जिसका उपयोग तब किया जाता है जब सॉफ्टवेयर ग्राहक को सौंपने के लिए तैयार होता है, इसे परीक्षण के अंतिम चरण को पास करना होगा, जिसमें उपयोगकर्ता सहभागिता और प्रतिक्रिया परीक्षण शामिल है। यह महत्वपूर्ण है क्योंकि, भले ही सॉफ्टवेयर उपयोगकर्ता की सभी ज़रूरतों को पूरा करता हो, अगर उपयोगकर्ता को उसके दिखने या कार्य करने के तरीके को नापसंद करने पर इसे अस्वीकार कर दिया जा सकता है।

अल्फा परीक्षण - डेवलपर टीम वास्तविक दुनिया की सेटिंग में सिस्टम को अपने पेस के माध्यम से रखकर अल्फा परीक्षण करती है। वे यह निर्धारित करने का प्रयास कर रहे हैं कि उपयोगकर्ता किसी सॉफ्टवेयर ऑपरेशन पर कैसे प्रतिक्रिया देगा और सिस्टम को इनपुट पर कैसे प्रतिक्रिया देनी चाहिए।

बीटा परीक्षण - सॉफ्टवेयर के आंतरिक रूप से पूरी तरह से परीक्षण के बाद, यह उपयोगकर्ताओं को केवल उनके उत्पादन वातावरण में परीक्षण उद्देश्यों के लिए दिया जाता है। यह अभी तैयार उत्पाद नहीं है। डेवलपर्स का अनुमान है कि इस स्तर पर, उपभोक्ता मामूली मुद्दों को लाएंगे जिन्हें उपस्थित होने के लिए अनदेखा किया गया था।

### प्रतिगमन परीक्षण

जब भी किसी सॉफ्टवेयर उत्पाद को नए कोड, सुविधा या कार्यक्षमता के साथ अपडेट किया जाता है, तो यह पता लगाने के लिए पूरी तरह से परीक्षण किया जाता है कि जोड़े गए कोड का कोई नकारात्मक प्रभाव तो नहीं है। इसे प्रतिगमन परीक्षण के रूप में जाना जाता है।

### 5.1.6 परीक्षण दस्तावेज़ीकरण

परीक्षण दस्तावेज विभिन्न चरणों में तैयार किए जाते हैं -

परीक्षण से पहले

टेस्ट केस जनरेशन के साथ टेस्टिंग शुरू होती है। संदर्भ के लिए निम्नलिखित दस्तावेजों की आवश्यकता है -

एसआरएस दस्तावेज	परीक्षण नीति दस्तावेज	परीक्षण रणनीति दस्तावेज	ट्रेसबिलिटी मैट्रिक्स दस्तावेज
<ul style="list-style-type: none"> <li>कार्यात्मक आवश्यकताएँ दस्तावेज</li> </ul>	<ul style="list-style-type: none"> <li>यह बताता है कि उत्पाद को जारी करने से पहले कितनी दूर परीक्षण किया जाता चाहिए।</li> </ul>	<ul style="list-style-type: none"> <li>इसमें परीक्षण टीम के विस्तृत पहलुओं, जिम्मेदारी मैट्रिक्स और परीक्षण पंचक और परीक्षण इंजीनियर के अधिकार/जिम्मेदारी का उल्लेख है।</li> </ul>	<ul style="list-style-type: none"> <li>यह एक ही एनवी में आवश्यकताएँ एकत्र करने के चरण से संबंधित एक दस्तावेज है। नई जरूरतें प्राप्त होने ही इस मैट्रिक्स को अपडेट किया जाता है। ये मैट्रिक्स मांग के स्रोत को लिपिबद्ध करने में परीक्षकों की सहायता करते हैं। उनका पालन दोनों दिशाओं में किया जा सकता है।</li> </ul>

चित्र 5.11 परीक्षण से पहले आवश्यक दस्तावेज

परीक्षण के दौरान

परीक्षण शुरू होने और किए जाने के दौरान निम्नलिखित दस्तावेजों की आवश्यकता हो सकती है:

टेस्ट केस दस्तावेज	विवरण परीक्षण	टेस्ट केस रिपोर्ट	परीक्षण लॉग
<p>इस दस्तावेज में आयोजित किए जाने के लिए आवश्यक परीक्षणों की सूची है। इसमें यूनिट टेस्ट प्लान, इंटीग्रेशन टेस्ट प्लान, सिस्टम टेस्ट प्लान और एक्सेप्टेड टेस्ट प्लान शामिल हैं।</p>	<p>यह दस्तावेज सभी परीक्षण मामलों और उन्हें निष्पादित करने की प्रक्रियाओं का विस्तृत विवरण है।</p>	<p>इस दस्तावेज में परीक्षण के परिणामस्वरूप परीक्षण मामले की रिपोर्ट है।</p>	<p>इस दस्तावेज में प्रत्येक परीक्षण मामले की रिपोर्ट के लिए परीक्षण लॉग हैं</p>

चित्र 5.12 परीक्षण के दौरान आवश्यक दस्तावेज

परीक्षण के बाद

परीक्षण के बाद निम्नलिखित दस्तावेज तैयार किए जा सकते हैं:

परीक्षण सारांश - यह परीक्षण सारांश सभी परीक्षण रिपोर्टों और लॉग का एक सामूहिक विश्लेषण है। यदि सॉफ्टवेयर लॉन्च करने के लिए तैयार है, तो यह सारांश और निष्कर्ष निकालता है। सॉफ्टवेयर को संस्करण नियंत्रण प्रणाली के तहत जारी किया जाता है यदि यह लॉन्च करने के लिए तैयार है।

## सारांश

- सत्यापन यह निर्धारित करने की प्रक्रिया है कि सॉफ्टवेयर उपयोगकर्ताओं की जरूरतों को पूरा करता है या नहीं। यह एसडीएलसी के बाद होता है। सत्यापन यह निर्धारित करता है कि सॉफ्टवेयर उन विशिष्टताओं को पूरा करता है या नहीं जिसके लिए इसे बनाया गया था।
- सत्यापन गारंटी देता है कि विकास में उत्पाद उपयोगकर्ताओं की जरूरतों को पूरा करता है।
- चिकित्सा उपकरण सॉफ्टवेयर में दोनों सॉफ्टवेयर शामिल होते हैं जिनका उपयोग किसी चिकित्सा उपकरण के घटक, भाग या सहायक उपकरण के रूप में किया जाता है और सॉफ्टवेयर जो अपने आप में एक चिकित्सा उपकरण है।
- उत्पादन सॉफ्टवेयर एफडीए-विनियमित उत्पादों के निर्माण में प्रयुक्त सॉफ्टवेयर और प्रणालियों को संदर्भित करता है।
- उत्पादों की गुणवत्ता सुनिश्चित करने के लिए सिस्टम में गुणवत्ता प्रबंधन सॉफ्टवेयर और सिस्टम का उपयोग किया जाता है।
- FDA-विनियमित रिकॉर्ड के लिए सॉफ्टवेयर को FDA-आवश्यक रिकॉर्ड बनाने, संशोधित करने, बनाए रखने, संग्रह करने, पुनर्प्राप्त करने या संचारित करने के लिए उपयोग किए जाने वाले सॉफ्टवेयर के रूप में परिभाषित किया गया है।
- सॉफ्टवेयर सत्यापन किसी भी सॉफ्टवेयर की खामियों को खोजने के लिए समीक्षा करने की प्रक्रिया है। सत्यापन यह सुनिश्चित करने की प्रक्रिया है कि सॉफ्टवेयर बिना किसी दोष के अपने उद्देश्यों को पूरा करता है। यह निर्धारित करने की प्रक्रिया है कि विकसित उत्पाद सही है या नहीं।
- निरीक्षण सत्यापन का सबसे संरचित और औपचारिक तरीका है और इसे आम तौर पर निरीक्षण के रूप में संदर्भित किया जाता है। निष्पक्ष मॉडरेटर के साथ तीन से छह लोगों का एक समूह बनाया जाता है।
- सॉफ्टवेयर परीक्षण उपयोगकर्ताओं और सिस्टम विनिर्देशों से एकत्र की गई आवश्यकताओं के खिलाफ सॉफ्टवेयर का मूल्यांकन है।
- गुणवत्ता आश्वासन त्रुटि के पुनरुत्पादन के लिए परिदृश्यों और त्रुटियों की एक विस्तृत श्रृंखला खोजने में आपकी सहायता कर सकता है। यह बहुत आसान है, और डेवलपर्स को इसे जल्द से जल्द ठीक करना चाहिए।
- एक परीक्षण मामला एक दस्तावेज है जिसमें परीक्षण डेटा का एक सेट, पूर्व शर्त, अपेक्षित परिणाम, और एक निश्चित आवश्यकता के अनुपालन को सत्यापित करने के लिए एक विशिष्ट परीक्षण परिदृश्य के लिए बनाई गई पोस्ट शर्तें शामिल हैं।
- ब्लैक-बॉक्स परीक्षण तब होता है जब वास्तविक कार्यान्वयन को ध्यान में रखे बिना कार्यक्षमता का परीक्षण किया जाता है।
- व्हाइट-बॉक्स परीक्षण: यह कोड दक्षता या संरचना में सुधार के लिए कार्यक्रम और इसके कार्यान्वयन का परीक्षण करने के लिए आयोजित किया जाता है। इसे 'संरचनात्मक' परीक्षण के रूप में भी जाना जाता है।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

व्हाइट-बॉक्स परीक्षण की तकनीकें क्या हैं?

परीक्षण दस्तावेज़ क्या है?

परीक्षण स्तरों का वर्णन करें।

परीक्षण योजना पहचानकर्ताओं की तालिका की व्याख्या और वर्णन करें।

रिक्त स्थान भरें:

\_\_\_\_\_ उपयोगकर्ताओं और सिस्टम विनिर्देशों से एकत्र की गई आवश्यकताओं के विरुद्ध सॉफ्टवेयर का मूल्यांकन है।

ख) सत्यापन प्रक्रिया के दौरान खामियों की पहचान करने के लिए दस्तावेज़ों या कार्यक्रमों/सॉफ्टवेयर की समीक्षा करने का सबसे सरल और सबसे अनौपचारिक तरीका है।

\_\_\_\_\_ किसी भी सॉफ्टवेयर की खामियों को खोजने के लिए समीक्षा करने की प्रक्रिया है।

एमसीक्यू प्रश्न:

निम्नलिखित में से कौन परीक्षण दस्तावेज़ का हिस्सा नहीं है?

टेस्ट केस

आवश्यकताएँ ट्रेसिबिलिटी मैट्रिक्स [RTM]

टेस्ट रणनीति

परियोजना की शुरुआत नोट [पिन]

परीक्षण को परिभाषित करने के लिए किस शब्द का प्रयोग किया जाता है?

त्रुटियों को खोजने के लिए सुपुर्दगी का मूल्यांकन

टूटा हुआ कोड ढूँढना

सभी परियोजनाओं का एक चरण

इनमें से कोई भी नहीं



## 6. प्रोग्रामिंग और सॉफ्टवेयर डेवलपमेंट के लिए एल्गोरिथ्म



IT - ITeS SSC  
NASSCOM

यूनिट 6.1 सॉफ्टवेयर प्रोग्रामिंग और एल्गोरिथ्म विकास



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. विनिर्देशों के अनुसार सॉफ्टवेयर निर्माण करने में सहायता के लिए उपयुक्त मानकों को लागू करने में
2. सॉफ्टवेयर डेवलपमेंट की जरूरतों और परिवर्तनों की पहचान करने में

## यूनिट 6.1: सॉफ्टवेयर प्रोग्रामिंग और एल्गोरिथम डेवलपमेंट

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

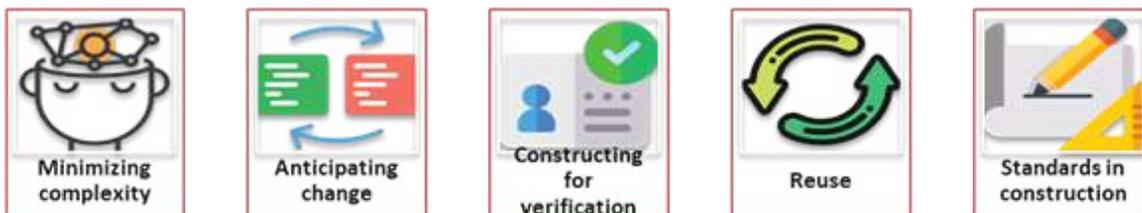
1. कम्प्यूटेशनल समस्याओं को हल करने में शामिल चरणों को सूचीबद्ध करने में
2. डेटा प्रवाह आरेखों के नुकसान को सूचीबद्ध करने में
3. सॉफ्टवेयर प्रोग्रामिंग के लिए एल्गोरिथम विकास की प्रक्रिया की पहचान करने में

### 6.1.1 विनिर्देशों के अनुसार सॉफ्टवेयर निर्माण के लिए उपयुक्त मानक

सॉफ्टवेयर निर्माण कोडिंग, सत्यापन, इकाई परीक्षण, एकीकरण परीक्षण और डिबगिंग के संयोजन के माध्यम से काम करने, सार्थक सॉफ्टवेयर की विस्तृत रचना है। सॉफ्टवेयर निर्माण सॉफ्टवेयर इंजीनियरिंग, विशेष रूप से सॉफ्टवेयर डिजाइन और परीक्षण के अन्य सभी पहलुओं के साथ दृढ़ता से जुड़ा हुआ है। यह इस तथ्य के कारण है कि सॉफ्टवेयर विकास प्रक्रिया महत्वपूर्ण सॉफ्टवेयर डिजाइन और परीक्षण गतिविधि को पूरा करती है। यह डिजाइन के इनपुट में से एक के रूप में डिजाइन के आउटपुट का उपयोग करता है, दोनों डिजाइन और परीक्षण गतिविधियों के साथ। डिजाइन, निर्माण और परीक्षण (यदि कोई हो) के बीच सटीक सीमाएं एक परियोजना में उपयोग किए जाने वाले सॉफ्टवेयर जीवन चक्र प्रक्रियाओं के आधार पर भिन्न होंगी।

चूंकि निर्माण से पहले कुछ विस्तृत डिजाइन कार्य हो जाते हैं, इसलिए निर्माण प्रक्रिया के दौरान अधिकांश डिजाइन कार्य किए जाते हैं। नतीजतन, सॉफ्टवेयर निर्माण और सॉफ्टवेयर डिजाइन अटूट रूप से जुड़े हुए हैं। विकास प्रक्रिया के दौरान, सॉफ्टवेयर इंजीनियर की यूनिट-टेस्ट और उनके काम को एकीकृत करें। इस प्रकार, सॉफ्टवेयर विकास सॉफ्टवेयर परीक्षण से भी अटूट रूप से जुड़ा हुआ है। सॉफ्टवेयर विकास आम तौर पर सबसे अधिक कॉन्फिगरेशन आइटम उत्पन्न करता है जिसे एक सॉफ्टवेयर प्रोजेक्ट (स्रोत फाइलों, सामग्री, परीक्षण मामलों, और इसी तरह) में प्रबंधित किया जाना चाहिए। नतीजतन, सॉफ्टवेयर विकास सॉफ्टवेयर कॉन्फिगरेशन प्रबंधन से अटूट रूप से जुड़ा हुआ है।

निम्नलिखित सॉफ्टवेयर विकास के मूल बुनियादी बातें/मानक हैं:



चित्र 6.1 सॉफ्टवेयर निर्माण के मूल सिद्धांत

जटिलता को कम करना: अधिकांश लोगों में अपनी कामकाजी यादों में जटिल संरचनाओं और सूचनाओं को बनाए रखने की सीमित क्षमता होती है, विशेष रूप से लंबे समय तक। इसका एक महत्वपूर्ण प्रभाव है कि कैसे लोग कंप्यूटर के लिए अपने इरादों का संचार करते हैं, जिसके परिणामस्वरूप सॉफ्टवेयर विकास में सबसे शक्तिशाली बलों में से एक है: जटिलता को कम करना। जटिलता को कम करने की आवश्यकता सॉफ्टवेयर विकास के लगभग हर पहलू पर लागू होती है, लेकिन यह सॉफ्टवेयर परीक्षण में विशेष रूप से महत्वपूर्ण है। चतुर कोड निर्माण पर सरल और पठनीय कोड निर्माण पर जोर देकर सॉफ्टवेयर विकास जटिलता कम हो जाती है।

परिवर्तन की आशंका: अधिकांश सॉफ्टवेयर समय के साथ विकसित होंगे, और परिवर्तन की प्रत्याशा सॉफ्टवेयर विकास के कई पहलुओं को चलाता है; उन वातावरणों में परिवर्तन जिसमें सॉफ्टवेयर संचालित होता है, सॉफ्टवेयर पर विभिन्न प्रकार का प्रभाव भी होता है। परिवर्तन की आशंका सॉफ्टवेयर इंजीनियरों को एक्स्टेंसिबल सॉफ्टवेयर विकसित करने में

सहायता करती है, जो उन्हें इसकी अंतर्निहित संरचना को बाधित किए बिना एक सॉफ्टवेयर उत्पाद में सुधार करने की अनुमति देता है। कई विशिष्ट तकनीकें परिवर्तन की आशंका में सहायता करती हैं।

3. सत्यापन के लिए निर्माण: सत्यापन के लिए निर्माण सॉफ्टवेयर को इस तरह से विकसित करना शामिल है कि सॉफ्टवेयर लिखने वाले सॉफ्टवेयर इंजीनियरों के साथ-साथ परीक्षकों और उपयोगकर्ताओं द्वारा स्वतंत्र परीक्षण और परिचालन गतिविधियों के दौरान दोषों को आसानी से पहचाना जा सके। कोड समीक्षा और इकाई परीक्षण का समर्थन करने के लिए निम्नलिखित कोडिंग मानकों, स्वचालित परीक्षण का समर्थन करने के लिए कोड का आयोजन, और जटिल या कठिन-से-समझने वाली भाषा संरचनाओं के उपयोग को सीमित करना कुछ विशिष्ट तकनीकें हैं जो सत्यापन के लिए निर्माण का समर्थन करती हैं।

4. पुनः उपयोग: शब्द "पुनः उपयोग" विभिन्न समस्याओं को हल करने के लिए मौजूदा परिसंपत्तियों के उपयोग को संदर्भित करता है। सॉफ्टवेयर विकास में आम तौर पर पुनः उपयोग की जाने वाली संपत्तियों में पुस्तकालय, मॉड्यूल, घटक, स्रोत कोड और वाणिज्यिक ऑफ-द-शेल्फ (सीओटीएस) संपत्तियां शामिल हैं। पुनः उपयोग सबसे अच्छा काम करता है जब एक अच्छी तरह से परिभाषित, दोहराने योग्य प्रक्रिया के अनुसार व्यवस्थित रूप से किया जाता है। व्यवस्थित पुनः उपयोग से सॉफ्टवेयर उत्पादकता, गुणवत्ता और लागत में महत्वपूर्ण लाभ हो सकता है। पुनः उपयोग के लिए निर्माण और पुनः उपयोग के साथ निर्माण, पुनः उपयोग के दो निकट से संबंधित पहलू हैं। पहला पुनः प्रयोज्य सॉफ्टवेयर परिसंपत्तियों के निर्माण को संदर्भित करता है, जबकि बाद वाला एक नए समाधान के विकास में सॉफ्टवेयर परिसंपत्तियों के पुनः उपयोग को संदर्भित करता है। पुनः उपयोग अक्सर परियोजना की सीमाओं को पार करता है, जिसका अर्थ है कि पुनः उपयोग की गई संपत्ति अन्य परियोजनाओं या संगठनों में बनाई जा सकती है।

5. निर्माण में मानक: निर्माण के दौरान बाहरी या आंतरिक विकास मानकों का उपयोग दक्षता, गुणवत्ता और लागत के लिए परियोजना के लक्ष्यों को पूरा करने में सहायता करता है। स्वीकार्य प्रोग्रामिंग भाषा उपसमुच्चय और उपयोग मानक, विशेष रूप से, उच्च सुरक्षा प्राप्त करने में महत्वपूर्ण सहायक हैं। निर्माण के मुद्दों पर सीधा प्रभाव डालने वाले मानकों में शामिल हैं:

संचार के तरीके (उदाहरण के लिए, दस्तावेज़ प्रारूपों और सामग्री के लिए मानक)

प्रोग्रामिंग भाषाएं (उदाहरण के लिए, जावा और सी ++ भाषा मानक) \*कोडिंग मानक (उदाहरण के लिए, नामकरण सम्मेलनों, लेआउट और इंडेंटेशन के लिए मानक)

प्लेटफॉर्म (उदाहरण के लिए, ऑपरेटिंग सिस्टम कॉल के लिए इंटरफ़ेस मानक)

उपकरण (उदाहरण के लिए, UML (एकीकृत मॉडलिंग भाषा) जैसे आरेखीय संकेतन मानक।

## 6.1.2 कम्प्यूटेशनल समस्याओं में शामिल कदम

एक कम्प्यूटेशनल समस्या वह है जिसे कंप्यूटर हल कर सकता है या एक जिसे कंप्यूटर जवाब दे सकता है। फैक्टरिंग समस्या पर विचार करें: "एक सकारात्मक पूर्णांक  $n$  दिए गए  $n$  का एक गैर-अभाज्य अभाज्य गुणनखंड खोजें।"

कंप्यूटर द्वारा एक कम्प्यूटेशनल समस्या को चरण दर चरण हल किया जा सकता है। इन समस्याओं में आमतौर पर एक अच्छी तरह से परिभाषित इनपुट, बाधाएं और शर्तें होती हैं जिन्हें आउटपुट को पूरा करना चाहिए। एक कम्प्यूटेशनल समस्या को उदाहरणों या मामलों के संग्रह के रूप में माना जा सकता है, जिनमें से प्रत्येक का अपना समाधान होता है, जो खाली हो भी सकता है और नहीं भी। उदाहरण के लिए, फैक्टरिंग समस्या में, उदाहरण पूर्णांक  $n$  होते हैं, और समाधान अभाज्य संख्याएँ  $p$  होते हैं जो  $n$  के गैर-तुच्छ अभाज्य गुणनखंडों का वर्णन करते हैं।



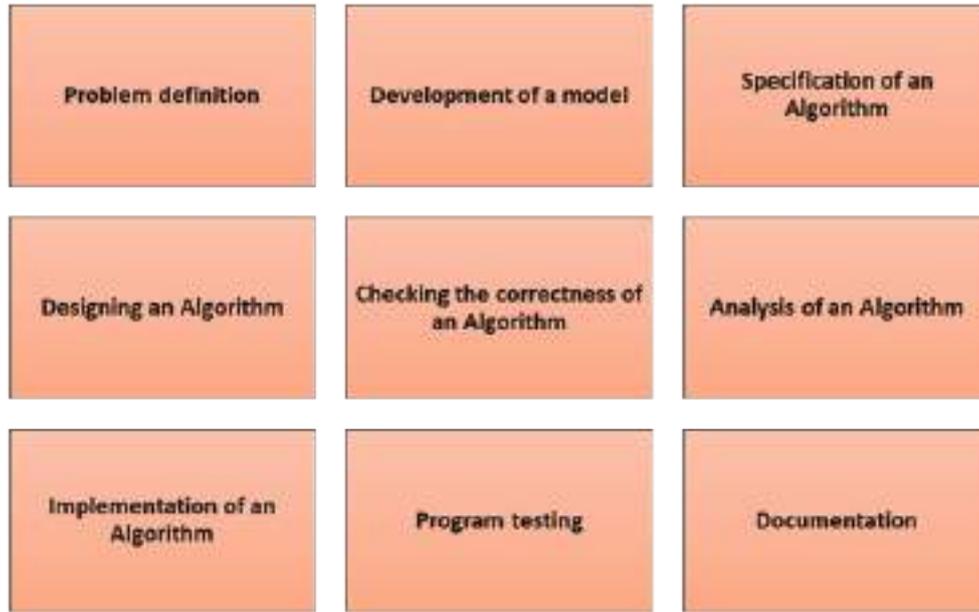
चित्र 6.2 कम्प्यूटेशनल समस्या

कम्प्यूटेशनल समस्याओं के प्रकार:

	<p>एक निर्णय समस्या हां या ना में उत्तर देने वाली होती है। एक निर्णय समस्या है, उदाहरण के लिए, "एक संख्या <math>n</math> ही गई है, क्या <math>n</math> सम है?" कुछ निर्णय समस्याओं के लिए दूसरों की तुलना में अधिक धरणों की आवश्यकता होती है। उदाहरण के लिए, "दिया गया संख्या <math>n</math>, क्या <math>n</math> अभाज्य है?" किसी संख्या की प्रकृति की जांच करने की तुलना में अधिक धरणों की आवश्यकता होती है।</p>
	<p>एक खोज समस्या का एक समाधान होता है जिसमें एक या एक से अधिक मान होते हैं जो किसी ही गई शर्त को पूरा करते हैं। उदाहरण के लिए, हम जानचिच पर एक भौगोलिक स्थल से दूसरे स्थल पर पथ की गणना करना चाह सकते हैं।</p>
	<p>एक गिनती समस्या यह है जिसमें समाधान एक खोज समस्या के संबंधित समाधानों की संख्या है।</p>
	<p>एक अनुकूलन समस्या यह है जिसमें समाधान "सर्वश्रेष्ठ" संभव समाधान है, जिसमें "सर्वश्रेष्ठ" को विभिन्न तरीकों में परिभाषित किया गया है। उदाहरण के लिए, हम एक स्थल से दूसरे स्थल के लिए सबसे छोटा मार्ग खोजना चाह सकते हैं।</p>

चित्र 6.3 कम्प्यूटेशनल समस्याओं के प्रकार

डेटा संग्रह या डेटा सेट (संख्यात्मक, पाठ, ऑडियो, वीडियो, छवियों, या प्रतीकों या प्रतीकों) की पहचान और समानताएं, अंतर, या रुझानों की खोज करने के लिए विश्लेषण। जटिलता को कम करने के लिए अप्रासंगिक जानकारी को फ़िल्टर करना। यह समस्या को सुलझाने और कम्प्यूटेशनल समस्या की सामान्य समझ बनाने में मदद कर सकता है।



चित्र 6.4 प्रॉब्लम डेवलपमेंट कदम

### 6.1.3 सॉफ्टवेयर आवश्यकताएँ और परिवर्तन

यह व्यवसायों के लिए महत्वपूर्ण है क्योंकि यह उन्हें प्रतिस्पर्धियों से बाहर खड़े होने और अधिक प्रतिस्पर्धी बनने की अनुमति देता है। ग्राहक के अनुभव को बढ़ाना, अधिक सुविधा संपन्न और नवीन उत्पादों को बाजार में लाना, और सेटअप को सुरक्षित, अधिक उत्पादक और अधिक कुशल बनाना सॉफ्टवेयर विकास के सभी लाभ हैं। डिजिटलीकरण और ऑनलाइन सूचना भंडारण न केवल स्थान बचाते हैं बल्कि सूचनाओं को एकीकृत और केंद्रीकृत भी करते हैं, जिससे यह उन लोगों के लिए आसानी से सुलभ हो जाता है जिन्हें इसकी आवश्यकता होती है। घुसपैठियों के खिलाफ अपने डेटा की सुरक्षा करना भी आसान है। जैसे-जैसे कंपनी का विस्तार होगा, वैसे-वैसे डेटा का पूल, और इस डेटा को कुशलतापूर्वक संग्रहीत करना प्राथमिकता बन जाएगा। इस डेटा का उपयोग करने वालों को लगातार प्रदर्शन प्रदान करना जारी रखने के लिए, व्यवसायों को अपने डिजिटल संगठनों को विकसित और अपग्रेड करना होगा। डेटा विश्लेषण के लिए सॉफ्टवेयर विकास की भी आवश्यकता होती है। जब सही सॉफ्टवेयर के साथ जोड़ा जाता है, तो व्यवसाय अपने ग्राहकों के बीच रुझानों पर नज़र रखने के लिए दिन-प्रतिदिन के कार्यों से एकत्र किए गए डेटा का उपयोग कर सकते हैं।

व्यापार में सॉफ्टवेयर विकास का उद्देश्य:



अपने व्यवसाय को बढ़ावा दें:

सॉफ्टवेयर का विकास आपकी कंपनी को नई ऊंचाइयों पर ले जाता है। यह सभी को और लगभग कहीं भी एक कंप्यूटर या स्मार्टफोन का उपयोग करके इसे दिखाने के लिए ब्रांड को बढ़ावा देता है और फैलाता है।



बिक्री और सेवा को बढ़ाता है:

यह समझना महत्वपूर्ण है कि ग्राहक ब्रांड और उत्पादों के बारे में क्या सोचते हैं। यदि व्यवसाय उनकी राय जानना चाहते हैं और उन्हें सकारात्मक टिप्पणी छोड़ना चाहते हैं, तो उनके पास एक ऑनलाइन प्लेटफॉर्म होना चाहिए जो ग्राहकों को आसानी से उनसे संपर्क करने और उत्पादों और सेवाओं पर अपने विचार साझा करने की अनुमति देता है।



प्रत्यक्ष संचार:

केवल एक चीज जो व्यवसायों को ग्राहकों के साथ प्रत्यक्ष संचार में मदद कर सकती है, वह है सॉफ्टवेयर विकास। कोई अन्य रणनीति उनके ग्राहकों के साथ सीधे संचार करने में उनकी सहायता नहीं कर सकती है। यह ब्रांड जागृक्ता बढ़ाने का सबसे तेज तरीका है।



ग्राहक जुड़ाव में सुधार करता है:

हर कंपनी अपने ग्राहक आधार का विस्तार करना चाहती है। लेकिन एक कंपनी अपने ग्राहक आधार को कैसे बढ़ा सकती है? समाधान ऑनलाइन मार्केटिंग है। व्यवसायों को ऑनलाइन विपणन रणनीतियों को नियोजित करना चाहिए। वे एक मोबाइल ऐप या वेबसाइट की मदद से ग्राहक जुड़ाव बढ़ा सकते हैं, जिससे वे अपने प्रतियोगी के बजाय उनके पास लौट सकते हैं।

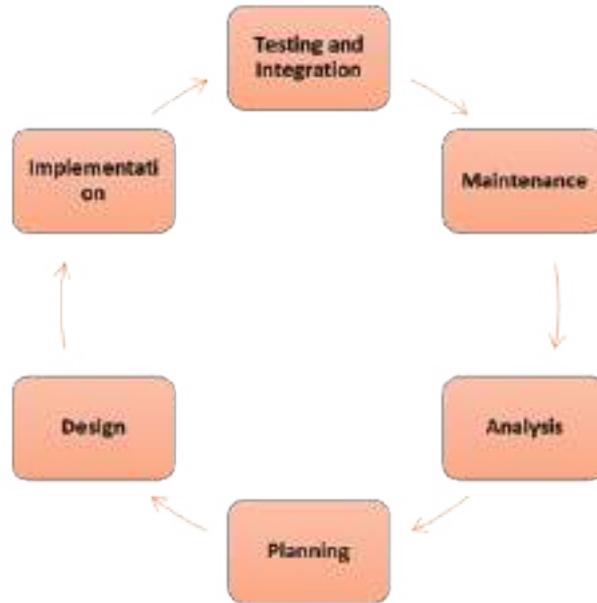


कंपनी के प्रचार में ऐड :

सॉफ्टवेयर विकास व्यवसायों को अपनी कंपनी के लिए ऑन-द-गो मार्केटिंग का संचालन करने में सक्षम बनाता है, किसी भी समय और किसी भी स्थान से किसी भी अतिरिक्त धन या समय को खर्च किए बिना उत्पादों और सेवाओं को बढ़ावा देता है। दुनिया भर के ग्राहक उनसे संपर्क कर सकते हैं।

चित्र 6.5 व्यवसाय में सॉफ्टवेयर डेवलपमेंट का उद्देश्य

सॉफ्टवेयर डेवेलोप करने में महत्वपूर्ण कदम:



चित्र 6.6 सॉफ्टवेयर डेवलपमेंट करने में महत्वपूर्ण कदम

### 6.1.4 डेटा प्रवाह आरेख

डेटा प्रवाह आरेख (DFD) एक सूचना प्रणाली में डेटा प्रवाह का चित्रमय चित्रण है। यह आने वाले डेटा प्रवाह, आउटगोइंग डेटा प्रवाह, और संग्रहीत डेटा प्रदर्शित करता है। DFD इस बात का कोई उल्लेख नहीं करता है कि सिस्टम के माध्यम से डेटा कैसे प्रवाहित होता है। इस तकनीक का उपयोग जटिल और कठिन-से-वर्णन प्रणालियों और प्रक्रियाओं को नेत्रहीन रूप से प्रस्तुत करने के लिए किया जाता है। डेटा प्रवाह आरेख एक प्रक्रिया या प्रणाली के माध्यम से डेटा के प्रवाह को दर्शाते हैं। इसमें डेटा इनपुट और आउटपुट, डेटा स्टोर और विभिन्न उप प्रक्रियाएं भी शामिल हैं जिनसे डेटा चलता है। DFD विभिन्न संस्थाओं और उनके संबंधों का वर्णन करने के लिए मानकीकृत संकेतन और प्रतीकों का उपयोग करता है। सिस्टम में खामियों की पहचान करना आसान होता है जब इसके सभी घटकों की कल्पना की जाती है। फिर एक मजबूत समाधान बनाने के लिए इन खामियों को दूर किया जाता है।



चित्र 6.7 डाटा फ्लो चित्र

DFD और फ्लोचार्ट के बीच एक महत्वपूर्ण अंतर है। फ्लोचार्ट प्रोग्राम मॉड्यूल के भीतर नियंत्रण प्रवाह को दर्शाता है। DFDs विभिन्न स्तरों पर एक सिस्टम में डेटा प्रवाह को दर्शाते हैं। DFD में कोई नियंत्रण या शाखा तत्व नहीं है।

डेटा प्रवाह आरेख के प्रकार

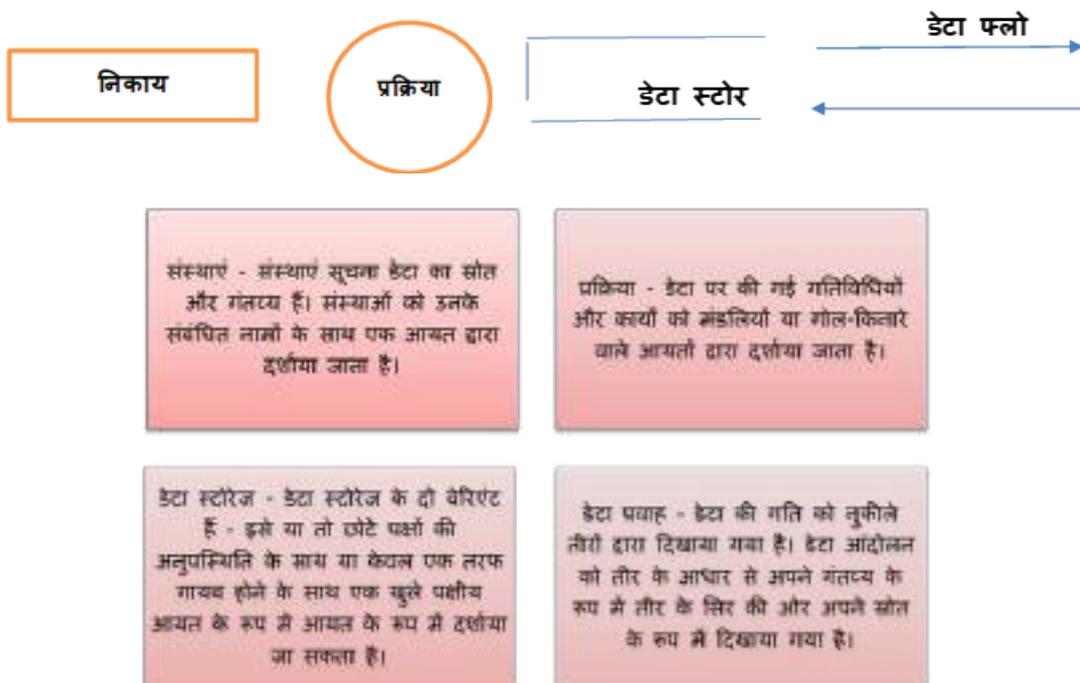
डेटा प्रवाह आरेखों को तार्किक या भौतिक के रूप में वर्गीकृत किया जाता है।

लॉजिकल डीएफडी - इस प्रकार का डीएफडी सिस्टम में सिस्टम प्रोसेस और डेटा फ्लो पर फोकस करता है। बैंकिंग सॉफ्टवेयर सिस्टम में, उदाहरण के लिए, विभिन्न संस्थाओं के बीच डेटा कैसे स्थानांतरित किया जाता है।

भौतिक डीएफडी - यह डीएफडी प्रकार दर्शाता है कि सिस्टम में डेटा प्रवाह कैसे लागू किया जाता है। यह अधिक विशिष्ट है और पूरा होने के करीब है।

डेटा प्रवाह आरेख के घटक

निम्नलिखित घटकों का उपयोग करते हुए, डीएफडी डेटा के स्रोत, गंतव्य, भंडारण और प्रवाह का प्रतिनिधित्व कर सकता है।



चित्र 6.8 डेटा घटक

डेटा प्रवाह आरेखों को डिजाइन करने के साथ संबंध महत्वपूर्ण अवधारणाएं:

सिस्टम के DFD मॉडल के निर्माण के विवरण में जाने से पहले, आइए DFD से जुड़ी कुछ महत्वपूर्ण अवधारणाओं पर चर्चा करें।

तुल्यकालिक और अतुल्यकालिक संचालन

यदि दो बुलबुले डेटा प्रवाह तीर द्वारा सीधे जुड़े हुए हैं, तो वे समकालिक होते हैं। इसका मतलब है कि वे एक ही दर से चलते हैं। यदि दो बुलबुले डेटा स्टोर के माध्यम से जुड़े हुए हैं, तो बुलबुले के संचालन की गति स्वतंत्र है। निर्माता बबल द्वारा उत्पन्न डेटा डेटा स्टोर में सहेजा जाता है। उपभोक्ता बबल किसी भी डेटा आइटम का उपभोग करने से पहले, निर्माता बबल उन्हें डेटा स्टोर में स्टोर कर सकता है।

### डेटा परिभाषा

निम्नलिखित डेटा परिभाषा ऑपरेटरों का उपयोग करके समग्र डेटा आइटम को आदिम डेटा आइटम के रूप में परिभाषित किया जा सकता है:

+ दो डेटा आइटम की संरचना को दर्शाता है, उदा।  $A+B$  डेटा  $A$  और  $B$  का प्रतिनिधित्व करता है।

[.] चयन का प्रतिनिधित्व करता है, यानी कोष्ठक में सूचीबद्ध डेटा आइटम में से कोई भी हो सकता है। उदाहरण के लिए,  $[A, B]$  या तो  $A$  होता है या  $B$  होता है।

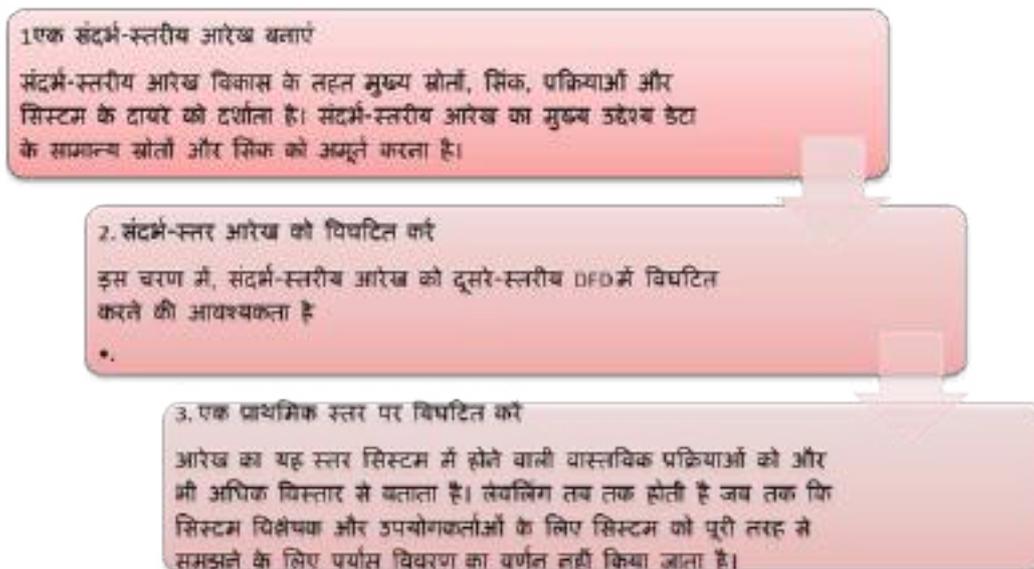
() ब्रैकेट के अंदर की सामग्री वैकल्पिक डेटा का प्रतिनिधित्व करती है जो दिखाई दे सकती है या नहीं। उदाहरण के लिए,  $A+(B)$  या तो  $A$  का प्रतिनिधित्व करता है या  $A+B$  होता है।

{ } Iterative डेटा परिभाषा का प्रतिनिधित्व करता है, उदाहरण के लिए,  $\{नाम\}$  5 पांच नाम डेटा का प्रतिनिधित्व करता है।  $\{नाम\}^*$  नाम डेटा के शून्य या अधिक उदाहरणों का प्रतिनिधित्व करता है।

= समानता का प्रतिनिधित्व करता है, उदाहरण के लिए,  $A = B+C$  का अर्थ है कि  $A$  का प्रतिनिधित्व करता है और  $C$ ।

/ \* \*/ कुछ भी/ \* और \*/// को एक टिप्पणी के रूप में माना जाता है।

एक डीएफडी निम्नलिखित चरणों द्वारा विकसित किया जाता है:



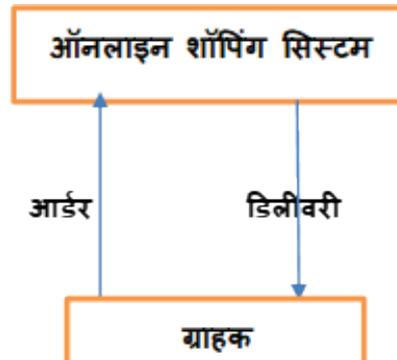
चित्र 6.9 एक डेटा प्रवाह आरेख विकसित करने में कदम

### डेटा प्रवाह आरेख के स्तर

ए) स्तर 0 - यह डीएफडी का उच्चतम अमूर्त स्तर है, और यह संपूर्ण सूचना प्रणाली को एक एकल आरेख के रूप में दर्शाता है जो सभी अंतर्निहित विवरणों को छुपाता है। कॉन्टेक्ट लेवल DFD, लेवल 0 DFD का दूसरा नाम है।

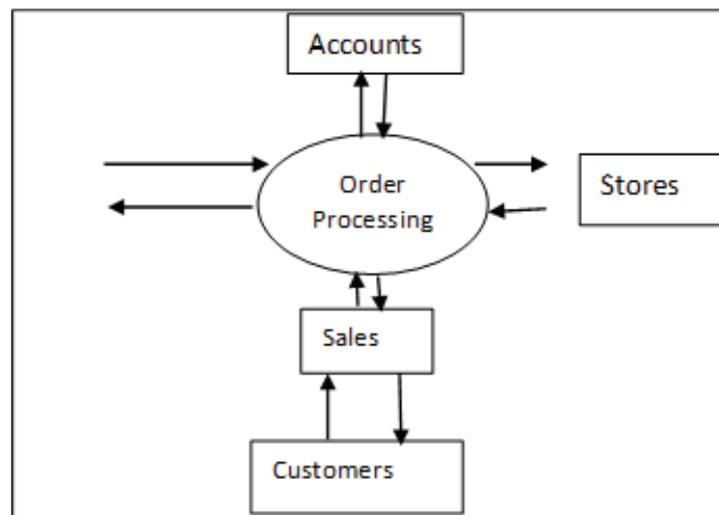
डेटा प्रवाह आरेख स्तर

स्तर 0 - यह DFD का उच्चतम अमूर्त स्तर है, और यह संपूर्ण सूचना प्रणाली को एक एकल आरेख के रूप में दर्शाता है जो सभी अंतर्निहित विवरणों को छुपाता है। संदर्भ स्तर DFDs स्तर 0 DFDs के लिए एक और नाम है।



चित्र 6.10 स्तर 0 DFD

स्तर 1 - स्तर 0 DFD को आगे स्तर 1 DFD में विभाजित किया गया है। स्तर 1 DFD में सिस्टम के मूल मॉड्यूल के साथ-साथ उनके बीच डेटा प्रवाह को भी दर्शाया गया है। स्तर 1 DFD में मौलिक प्रक्रियाओं और सूचना स्रोतों का भी उल्लेख है।



चित्र 6.11 Level 1 DFD

स्तर 2 - इस स्तर पर, DFD यह दर्शाता है कि इस स्तर पर स्तर 1 में उल्लिखित मॉड्यूल के भीतर डेटा कैसे बहता है। जब तक विनिर्देश का वांछित स्तर हासिल नहीं किया जाता है, तब तक उच्च-स्तरीय DFD को अधिक विशिष्ट निम्न-स्तरीय DFD में बदल दिया जा सकता है, जिसमें समझ के गहरे स्तर के साथ।

जैसा कि पहले कहा गया था, DFD विधि ऑब्जेक्ट-ओरिएंटेड विश्लेषण का एक घटक है और इसका व्यापक रूप से उपयोग किया जाता है। DFDs का उपयोग त्वरित और सरल परियोजना कोड विकास को प्रोत्साहित करता है। एफडी उनके कुछ सरल प्रतीकों के कारण सीखने के लिए सरल हैं (एक बार जब आप किसी विशेष डीएफडी मॉडल पर निर्णय लेते हैं)। DFDs को डिजाइन करने के लिए सिंटैक्स सीधा है, अंग्रेजी संज्ञाओं या संज्ञा-विशेषण-क्रिया निर्माणों पर निर्भर है।

#### विशिष्ट प्रक्रियाएँ

- यह एक प्रक्रिया, या बुलबुला, एक वृत्त, अंडाकार या आयत के रूप में दिखाता है।
- एक प्रक्रिया या बुलबुला एक प्रोग्राम फंक्शन का वर्णन करता है और डेटा प्रवाह आरेख विशिष्ट प्रोग्राम फंक्शन का वर्णन करने के लिए सरल नाम और संख्या प्रदान करते हैं।
- प्रक्रिया को क्रमांकित करके और प्रक्रिया के नामों को सरल रखते हुए, सिस्टम विश्लेषक विशिष्ट प्रक्रिया को आसानी से और सटीक रूप से संदर्भित कर सकता है।
- एक प्रक्रिया के उदाहरण हैं: मासिक वेतन की गणना करें, ध्याज दर की गणना करें और एक रिपोर्ट प्रिंट करें।

#### प्रवाह की स्पष्टता

- यह एक घुमावदार या सीधी तीर रेखा के रूप में एक प्रवाह दिखाता है।
- डेटा प्रवाह आरेख पर, तीर रेखा को डेटा या जानकारी का वर्णन करने के लिए लेबल किया जाता है जिसे स्थानांतरित या संसाधित किया जा रहा है।
- तीर रेखा का बिंदु स्रोत दिखाता है और तीर अगली प्रक्रिया, स्टोर या टर्मिनेटर को इंगित करता है।
- डेटा प्रवाह आरेख की सहायता से, एक सिस्टम विश्लेषक यह निर्धारित कर सकता है कि किसी सिस्टम में प्रक्रिया के लिए सभी आवश्यक इनपुट डेटा और आउटपुट डेटा है या नहीं।

#### उचित आंकड़ा भंडार

- यह एक स्टोर को दो समानांतर लाइनों के रूप में दिखाता है।
- एक स्टोर डेटा या जानकारी के लिए किसी भी प्रकार का स्टोरेज है और किसी भी स्वचालित डेटाबेस, फ़ाइल या मैन्युअल फ़ाइल सिस्टम को संदर्भित कर सकता है।
- एक डेटा फ्लो आरेख एक सिस्टम विश्लेषक को यह निर्धारित करने में सहायता कर सकता है कि क्या डेटा स्टोर में आवश्यक आउटपुट डेटा या जानकारी उत्पन्न करने के लिए आवश्यक इनपुट डेटा को संसाधित किया जा सकता है।
- एक डेटा प्रवाह आरेख यह निर्धारित करने में सहायता कर सकता है कि क्या डेटा स्टोर उत्पन्न आउटपुट जानकारी के लिए उपयुक्त है।

#### डेटा प्रवाह आरेख स्तर

- एक डेटा फ्लो आरेख में कई स्तर होते हैं जो एक सिस्टम एनालिस्ट को एक सिस्टम की समीक्षा करने में मदद करते हैं, प्रक्रिया द्वारा प्रक्रिया।
- एक स्तर 0 डेटा प्रवाह आरेख शीर्ष-स्तरीय आरेख है और इसमें केवल एक प्रक्रिया होती है।
- डेटा प्रवाह आरेख का अगला स्तर अधिक विवरण दिखाता है, और जैसे-जैसे स्तर अधिक होता है, अधिक विवरण दिखाया गया है।
- यह सिस्टम विश्लेषक को आसानी से विवरण को तोड़ने और किसी सिस्टम में होने वाली किसी भी समस्या को निर्धारित करने में मदद करता है।

चित्र 6.12 DFD के फायदे

डेटा प्रवाह आरेख की सीमाएँ

बड़ी प्रणालियों के लिए DFDs अनाड़ी, अनुवाद और पढ़ने के लिए मुश्किल हो सकते हैं, और बनाने के लिए समय लेने वाले हो सकते हैं। डेटा प्रवाह आरेख (DFD) प्रोग्रामर के लिए हैरान हो सकते हैं, लेकिन वे आवश्यक विवरण के बिना बेकार हैं। विभिन्न DFD मॉडल विभिन्न प्रतीकों (हलकों और आयतों, उदाहरण के लिए, संस्थाओं के लिए) का उपयोग करते हैं।

यह प्रोग्रामर को सिस्टम के विषय में थोड़ा भ्रमित करता है।

DFD का सबसे बड़ा दोष यह है कि इसे बनाने में बस एक लंबा समय लगता है, इसलिए कि विश्लेषक को इसे पूरा करने के लिए प्रबंधन से समर्थन प्राप्त नहीं हो सकता है।

शारीरिक विचारों को छोड़ दिया जाता है।

### 6.1.5 सॉफ्टवेयर प्रोग्रामिंग के लिए एल्गोरिथम विकास

एल्गोरिथम विकास, डिबगिंग और परीक्षण आमतौर पर MATLAB या C/C++ जैसे उच्च-स्तरीय प्रोग्रामिंग टूल का उपयोग करके किया जाता है। परीक्षण संकेतों पर सफल एल्गोरिथम विकास और सिमुलेशन के बाद एल्गोरिथम हार्डवेयर कार्यान्वयन के लिए तैयार है। विस्तृत डिजाइन का प्राथमिक लक्ष्य सिस्टम डिजाइन के दौरान वर्णित विभिन्न मॉड्यूल के लिए तर्क को निर्दिष्ट करना है।

तर्क को निर्दिष्ट करने से एक एल्गोरिथम के निर्माण की आवश्यकता होगी जो दिए गए विनिर्देशों को पूरा करेगा। आइए एल्गोरिथम या तर्क बनाने के लिए कुछ डिजाइन सिद्धांतों को देखें जो दिए गए विनिर्देशों को लागू करते हैं। "एल्गोरिथम" शब्द काफी व्यापक है और विषयों की एक विस्तृत श्रृंखला को कवर करता है।

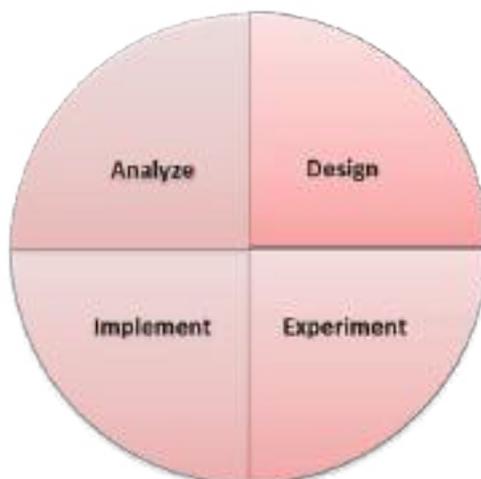
सॉफ्टवेयर के संदर्भ में, एक एल्गोरिथम एक समस्या को हल करने के लिए एक स्पष्ट प्रक्रिया है। एक प्रक्रिया अच्छी तरह से परिभाषित चरणों या संचालन का एक परिमित अनुक्रम है, प्रत्येक को पूरा करने के लिए स्मृति और समय की एक परिमित राशि की आवश्यकता होती है। इस परिभाषा में, हम मानते हैं कि प्रक्रिया समाप्ति एक आवश्यक संपत्ति है। अब हम "प्रक्रियाओं," "एल्गोरिथम, और" लॉजिक "शब्दों का उपयोग करेंगे। एक एल्गोरिथम बनाते समय, कई कदम उठाए जाने चाहिए।

एक समस्या कथन एल्गोरिथम के डिजाइन में पहला कदम है। जिस समस्या के लिए एक एल्गोरिथम डिजाइन किया जा रहा है, उसे ठीक से, स्पष्ट रूप से और अच्छी तरह से एल्गोरिथम डिजाइन करने के लिए जिम्मेदार व्यक्ति द्वारा कहा जाना चाहिए। विस्तृत डिजाइन के लिए समस्या कथन सिस्टम डिजाइन से लिया गया है। यही है, जब एक मॉड्यूल का विस्तृत डिजाइन शुरू होता है, तो समस्या कथन पहले से ही उपलब्ध है। अगला कदम समस्या का एक गणितीय मॉडल बनाना है।

मॉडलिंग में, समस्या के लिए सबसे उपयुक्त गणितीय संरचनाएं चुनी जानी चाहिए। अन्य समान समस्याओं को देखते हुए जो हल किए गए हैं, वे फायदेमंद हो सकते हैं। अधिकांश मॉडल समान समस्याओं से मॉडल लेने और वर्तमान समस्या को फिट करने के लिए उन्हें संशोधित करके बनाए जाते हैं। एल्गोरिथम का डिजाइन अगला कदम है।

इस चरण के दौरान डेटा संरचना और कार्यक्रम संरचना तय की जाती है। एल्गोरिथम डिजाइन किए जाने के बाद, इसकी शुद्धता को सत्यापित किया जाना चाहिए। एल्गोरिथम डिजाइन करने के लिए कोई स्पष्ट प्रक्रिया नहीं है। ऐसी प्रक्रिया का होना एल्गोरिथम विकास की समस्या को स्वचालित करने के बराबर होगा, जो वर्तमान विधियों से संभव नहीं है। हालांकि, मॉड्यूल के एल्गोरिथम को डिजाइन करने में डिजाइनर की सहायता के लिए कुछ अनुमान या विधियां प्रदान की जा सकती हैं।

एल्गोरिथम या मॉड्यूल लॉजिक को डिजाइन करने के लिए चरणबद्ध शोधन तकनीक सबसे अधिक इस्तेमाल की जाने वाली विधि है। चरणबद्ध शोधन तकनीक तर्क डिजाइन समस्या को क्रमिक विकास की अनुमति देते हुए, चरणों की एक श्रृंखला में विभाजित करती है। प्रक्रिया मॉड्यूल विनिर्देशों को एक एल्गोरिथम के एक सार विवरण में परिवर्तित करके शुरू होती है जिसमें कुछ सार विवरण होते हैं। प्रत्येक चरण में, पहले से विकसित एल्गोरिथम के एक या अधिक कथनों को अधिक विस्तृत निर्देशों में विघटित किया जाता है।



चित्र 6.13 एल्गोरिथम का डिजाइन और विश्लेषण

## एल्गोरिथ्म

एक एल्गोरिथ्म निर्देशों का एक सेट है, जिसे "प्रक्रिया" के रूप में भी जाना जाता है, जिसे एक विशिष्ट समस्या को हल करते समय पालन किया जाना चाहिए। जबकि शब्द तकनीकी रूप से परिभाषित नहीं किया गया है, यह लगभग हमेशा कंप्यूटर के साथ जुड़ा हुआ है क्योंकि कंप्यूटर-संसाधित एल्गोरिदम मनुष्यों की तुलना में बहुत बड़ी समस्याओं को हल कर सकते हैं। क्योंकि मानव इतिहास में किसी भी अन्य समय की तुलना में आधुनिक कंप्यूटिंग में एल्गोरिदम का उपयोग अधिक बार किया जाता है, एक क्षेत्र उनके डिजाइन, विश्लेषण और शोधन के आसपास विकसित हुआ है। एल्गोरिथ्म डिजाइन के क्षेत्र में एक मजबूत गणितीय पृष्ठभूमि की आवश्यकता होती है, जिसमें कंप्यूटर विज्ञान की डिग्री विशेष रूप से मांग की जाती है। यह अधिक (और अधिक परिष्कृत) एल्गोरिदम की मांग के रूप में अच्छी तरह से भुगतान करने वाले नौकरी के अवसरों की बढ़ती संख्या प्रदान करता है।

### एल्गोरिथ्म का वैचारिक डिजाइन:

एल्गोरिदम, अपने सबसे बुनियादी रूप से, केवल एक कार्य को पूरा करने के लिए आवश्यक निर्देशों का एक सेट है। सभी रिकॉर्ड किए गए इतिहास के लिए, एल्गोरिदम का विकास, हालांकि उन्हें आम तौर पर इस तरह से संदर्भित नहीं किया गया था, एक लोकप्रिय शौक और एक पेशेवर खोज रहा है। लोगों ने आधुनिक कंप्यूटर युग की सुबह से बहुत पहले अपने दैनिक कार्यों के बारे में कैसे जाना होगा, इस बारे में निर्धारित दिनचर्या की स्थापना की, अक्सर महत्वपूर्ण लक्ष्यों को पूरा करने के लिए कदमों की सूची लिखते हुए, कुछ महत्वपूर्ण को भूलने के जोखिम को कम करते हुए। यह, संक्षेप में, एक एल्गोरिथ्म क्या है। डिजाइनर उसी तरह से कम्प्यूटेशनल उद्देश्यों के लिए एल्गोरिदम के विकास को देखते हैं: वे एक समस्या के साथ शुरू करते हैं। फिर वे उन चरणों को रेखांकित करते हैं जिनके बारे में समस्या को हल करने के लिए आवश्यक होगा। अंत में, वे उन चरणों को पूरा करने के लिए गणितीय संचालन की एक श्रृंखला विकसित करते हैं।

### एल्गोरिथ्म डिजाइन

कम से कम समय और स्थान में एक समस्या को हल करने के लिए एक कुशल एल्गोरिथ्म बनाना एल्गोरिथ्म डिजाइन का एक महत्वपूर्ण पहलू है। एक समस्या को हल करने के लिए विभिन्न दृष्टिकोणों को लिया जा सकता है।

उनमें से कुछ समय की खपत के संदर्भ में कुशल हो सकते हैं, जबकि अन्य स्मृति कुशल हो सकते हैं। हालांकि, ध्यान रखें कि समय की खपत और मेमोरी उपयोग दोनों को एक ही समय में अनुकूलित नहीं किया जा सकता है। यदि हमें कम समय में चलाने के लिए एक एल्गोरिथ्म की आवश्यकता है, तो हमें अधिक मेमोरी में निवेश करना चाहिए, और यदि हमें कम मेमोरी में चलाने के लिए एक एल्गोरिथ्म की आवश्यकता है, तो हमें अधिक समय में निवेश करना चाहिए।

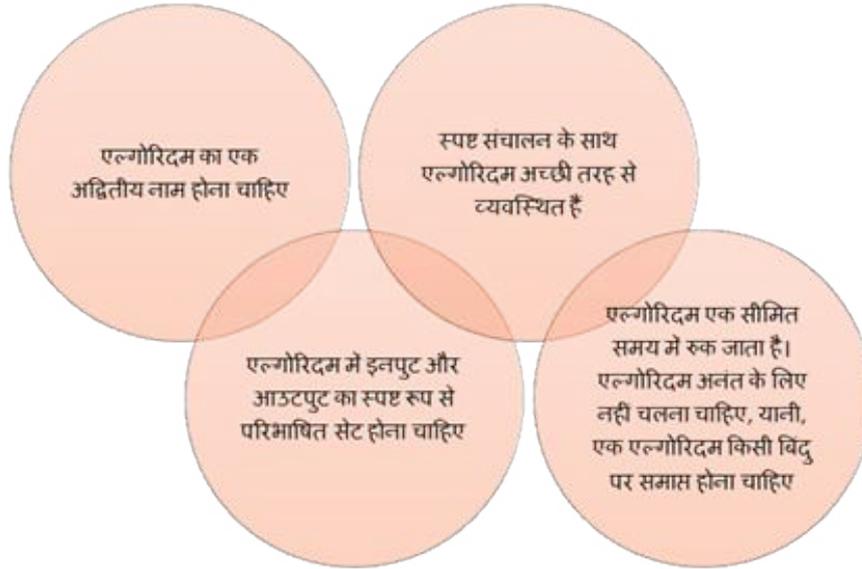
### समस्या विकास चरण:



चित्र 6.14 प्रॉब्लम डेवलपमेंट कदम

एल्गोरिथम की विशेषताएं:

एल्गोरिथम की मुख्य विशेषताएं इस प्रकार हैं -



चित्र 6.15 एल्गोरिथम के लक्षण

एल्गोरिथम की अनुवाद समस्याएं

अनुवादकों को आमतौर पर अपने काम में छह अलग-अलग समस्याग्रस्त क्षेत्रों से निपटना पड़ता है, चाहे वे तकनीकी दस्तावेजों का अनुवाद कर रहे हों या एक शपथ कथन। इनमें लेक्सिकल-सेमंटिक समस्याएं शामिल हैं; व्याकरण; वाक्य - विन्यास; बयानबाजी; और व्यावहारिक और सांस्कृतिक समस्याएं। प्रशासनिक मुद्दों, कंप्यूटर से संबंधित समस्याओं और तनाव का उल्लेख नहीं करना।

#### लेक्सिकल-सेमंटिक समस्याएं

•शब्दकोशों, शब्दावलीयों, शब्दावली बैंकों और विशेषज्ञों से परामर्श करके लेक्सिकल-सिमेंटिक समस्याओं का समाधान किया जा सकता है। इन समस्याओं में शब्दावली विकल्प, लवविज्ञान, शब्दार्थ अंतराल, प्रासंगिक पर्यायवाची शब्द और विलोम शामिल हैं (ये पॉलीसेमिक इकाइयों को प्रभावित करते हैं: समानार्थक शब्द और विलोम केवल एक स्वीकृति के उद्देश्य से होते हैं जो यह निर्धारित करने के लिए संदर्भ पर निर्भर करता है कि कौन सा अर्थ सही है), शब्दार्थ सन्निकितता (एक संगति प्रक्रिया) जो दो या दो से अधिक शब्दों के लिए सामान्य सिमेंटिक विशेषताओं की पहचान करके काम करता है) और लेक्सिकल नेटवर्क।

#### व्याकरण संबंधी समस्याएं

•व्याकरणिक समस्याओं में शामिल हैं, उदाहरण के लिए, अस्थायीता, पहलू के प्रश्न (उपस्थिति इंगित करती है कि प्रक्रिया का प्रतिनिधित्व कैसे किया जाता है या क्रिया द्वारा उसके विकास के दृष्टिकोण से क्रिया द्वारा व्यक्त किया जाता है, जैसा कि समय के विपरीत है), सर्वनाम, और क्या बनाता है विषय सर्वनाम को स्पष्ट करें या नहीं।

#### वाक्यविन्यास समस्याएं

•वाक्य-रचना संबंधी समस्याएं वाक्यात्मक समानताएं, निष्क्रिय आवाज की दिशा, फोकस (जिस दृष्टिकोण से एक कहानी का आयोजन किया जाता है), या यहां तक कि भाषण के अलंकारिक आंकड़े, जैसे हाइपरवेटन (भाषण के प्राकृतिक क्रम का उलटा) में उत्पन्न हो सकते हैं। ) या अनाफोरा (किसी पंक्ति या वाक्यांश की शुरुआत में किसी शब्द या खंड की पुनरावृत्ति)।

#### बयानवाजी की समस्या

•बयानवाजी की समस्याएं विचार के आंकड़ों की पहचान और मनोरंजन से संबंधित हैं (तुलना, रूपक, रूपक, मेटोमि, सिकडोचे, ऑक्सीमोरोन, विरोधाभास, आदि) और डिक्शन।

#### व्यावहारिक समस्याएं: एक विपणन अनुवाद का एक उदाहरण

•व्यावहारिक समस्याएं "आप" का उपयोग करके पते के औपचारिक और अनौपचारिक तरीकों में अंतर के साथ उत्पन्न होती हैं, साथ ही साथ मुहावरेदार वाक्यांशों, बातों, विडंबना, हास्य और व्यंग्य। इन कठिनाइयों में अन्य चुनौतियां भी शामिल हो सकती हैं: उदाहरण के लिए, अंग्रेजी से फ्रेंच में एक विपणन पाठ के अनुवाद में, विशेष रूप से व्यक्तिगत सर्वनाम "आप" के अनुवाद के साथ। अनुवादक को यह तय करना होगा कि औपचारिक या अनौपचारिक "आप" अधिक उपयुक्त है, एक निर्णय जो हमेशा स्पष्ट नहीं होता है।

#### सांस्कृतिक मुद्दे: एक वित्तीय अनुवाद का एक उदाहरण

•सांस्कृतिक संदर्भ सांस्कृतिक संदर्भों के बीच अंतर से उत्पन्न हो सकते हैं, जैसे कि भोजन, त्योहारों और सांस्कृतिक अर्थों के नाम, सामान्य रूप से। अनुवादक लक्षित संस्कृति में अनुवाद को सही ढंग से अनुकूलित करने के लिए भाषा स्थानीयकरण का उपयोग करेगा। एक बहुत ही सरल उदाहरण एक वित्तीय अनुवाद है जिसमें तिथियां शामिल हैं। यदि पाठ अंग्रेजी में है, तो यह सबसे अधिक संभावना है, लेकिन पूरी तरह से निश्चित नहीं है, कि 05/06/2015 का मतलब 5 जून होगा। हालांकि, जैसा कि सभी जानते हैं, किसी अन्य भाषा में एक ही अनुक्रम 6 मई को संदर्भित करता है।

चित्र 6.16 एल्गोरिथम की अनुवाद समस्याएं

क्रमिक शोधन प्रक्रिया तब समाप्त होती है जब सभी निर्देशों को आसानी से प्रोग्रामिंग भाषा के बयानों में परिवर्तित करने के लिए पर्याप्त सटीक होता है। शोधन प्रक्रिया के दौरान डेटा और निर्देश दोनों को परिष्कृत किया जाना चाहिए। शोधन के लिए अंगूठे का एक नियम यह है कि प्रत्येक चरण में अपघटन की मात्रा प्रबंधनीय होनी चाहिए और एक या दो डिजाइन निर्णयों का प्रतिनिधित्व करना चाहिए। सामान्य तौर पर, विस्तृत डिजाइन उन भाषाओं का उपयोग करके निर्दिष्ट किया जाता है जिनमें बाहरी संरचनाओं (जैसे लूप, सशर्त, आदि) जैसी औपचारिक प्रोग्रामिंग भाषा विशेषताएं होती हैं, लेकिन आंतरिक विवरण के लिए एक फ्रीर प्रारूप। यह डिजाइनर को प्रोग्रामिंग भाषा में इसके प्रतिनिधित्व के बजाय तर्क पर ध्यान केंद्रित करने में सक्षम बनाता है।

## सारांश



- एक विकास वातावरण, जिसे एक एकीकृत विकास वातावरण (IDE) के रूप में भी जाना जाता है, विकास उपकरणों के एक सेट को एकीकृत करके प्रोग्रामर को व्यापक सॉफ्टवेयर विकास सुविधाएं प्रदान करता है। एक कम्प्यूटेशनल समस्या यह है कि एक कंप्यूटर चरण दर चरण को हल कर सकता है। इन समस्याओं में आमतौर पर एक अच्छी तरह से परिभाषित इनपुट, बाधाएं और स्थितियां होती हैं जिन्हें आउटपुट द्वारा पूरा किया जाना चाहिए। एक कम्प्यूटेशनल समस्या को उदाहरणों या मामलों के संग्रह के रूप में माना जा सकता है, प्रत्येक अपने स्वयं के, संभवतः खाली, समाधानों के सेट के साथ।
- ऐसे कई चरण हैं जो एक एल्गोरिथम विकसित करते समय प्रदर्शन करना है। एल्गोरिथम के डिजाइन में शुरुआती कदम समस्या का एक बयान है। जिस समस्या के लिए एक एल्गोरिथम तैयार किया जा रहा है, उसे ठीक से और स्पष्ट रूप से कहा जाना चाहिए और एल्गोरिथम को डिजाइन करने के लिए जिम्मेदार व्यक्ति द्वारा ठीक से समझा जाना चाहिए।
- एल्गोरिथम विकास, डिबगिंग और परीक्षण आमतौर पर उच्च-स्तरीय प्रोग्रामिंग टूल जैसे कि MATLAB या C/C++ के साथ पूरा किया जाता है। परीक्षण संकेतों पर सफल एल्गोरिथम विकास और सिमुलेशन के बाद, एल्गोरिथम हार्डवेयर कार्यान्वयन के लिए तैयार है।
- उपयुक्त डेटा स्टोर: एक डेटा प्रवाह आरेख एक स्टोर को दो समानांतर लाइनों के रूप में दिखाता है। एक स्टोर डेटा या जानकारी के लिए किसी भी प्रकार का स्टोरेज है और किसी भी स्वचालित डेटाबेस, फ़ाइल या मैनुअल फ़ाइल सिस्टम को संदर्भित कर सकता है। एक डेटा फ्लो आरेख एक सिस्टम विश्लेषक को यह निर्धारित करने में सहायता कर सकता है कि क्या डेटा स्टोर में आवश्यक आउटपुट डेटा या जानकारी उत्पन्न करने के लिए आवश्यक इनपुट डेटा को संसाधित किया जा सकता है। एक डेटा प्रवाह आरेख यह निर्धारित करने में सहायता कर सकता है कि क्या डेटा स्टोर उत्पन्न आउटपुट जानकारी के लिए उपयुक्त है।
- विशिष्ट प्रक्रियाएं: एक डेटा प्रवाह आरेख एक प्रक्रिया, या बुलबुले को एक सर्कल, अंडाकार या आयत के रूप में दिखाता है। एक प्रक्रिया या बुलबुला एक कार्यक्रम फ़ंक्शन का वर्णन करता है और डेटा प्रवाह आरेख विशिष्ट कार्यक्रम फ़ंक्शन का वर्णन करने के लिए सरल नाम और संख्या प्रदान करते हैं। प्रक्रिया को नंबर देने और प्रक्रिया के नाम को सरल रखने से, सिस्टम विश्लेषक विशिष्ट प्रक्रिया को आसानी से और सटीक रूप से संदर्भित कर सकते हैं। एक प्रक्रिया के उदाहरण हैं: मासिक वेतन

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

DFD क्या है?

DFD के फायदे और नुकसान क्या हैं?

एल्गोरिथ्म विकास की व्याख्या करें।

डेटा परिभाषा की व्याख्या करें।

MCQ प्रश्न:

एक डेटा प्रवाह आरेख में जो निम्नलिखित प्रक्रिया में से सबसे विस्तृत स्तर पर है?

आंकड़ा प्रवाह

इंटरफेस

कार्यात्मक आदिम

परिवर्तन विवरण

ii। निम्नलिखित में से किस मॉडल का उपयोग यह दिखाने के लिए किया जाता है कि डेटा प्रसंस्करण चरणों के अनुक्रम के माध्यम से कैसे बहता है?

ऑब्जेक्ट्स मॉडल

सिस्टम मॉडल

शब्दार्थ डेटा मॉडल

आंकड़ा प्रवाह मॉडल

iii। निम्नलिखित में से कौन गति में डेटा प्रवाह की पहचान करता है?

सर्किल

तीर

खुले हुई बक्से

वर्ग

रिक्त स्थान भरें:

एक \_\_\_\_\_, निर्देशों की एक श्रृंखला, जिसे अक्सर एक "प्रक्रिया" के रूप में संदर्भित किया जाता है, जिसका पालन किसी विशेष समस्या को हल करते समय किया जाता है।

एक \_\_\_\_\_ एक प्रक्रिया, या बुलबुला, एक सर्कल, अंडाकार, या आयत के रूप में।

अनुवादकों को आमतौर पर अपने काम में छह अलग-अलग समस्याग्रस्त क्षेत्रों से निपटना पड़ता है, चाहे वे \_\_\_\_\_ या एक \_\_\_\_\_ हों।





## 7. सॉफ्टवेयर अनुप्रयोगों की डिजाइन



IT - ITeS SSC  
NASSCOM

यूनिट 7.1 सॉफ्टवेयर डेवलपमेंट लाइफ साईकल (चक्र)

यूनिट 7.2 सॉफ्टवेयर डेवलपमेंट प्रक्रिया के तत्व



## मुख्य सीख



इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. विभिन्न सॉफ्टवेयर परीक्षण पद्धति का मूल्यांकन करें और तैनात (डिप्लॉय) करने के लिए सही की पहचान करने में
2. पहले से निर्मित उत्पादों या सेवाओं के लिए सॉफ्टवेयर डिजाइन का विश्लेषण करने में

## यूनिट 7.1: सॉफ्टवेयर डेवलपमेंट लाइफ साइकल (चक्र)

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

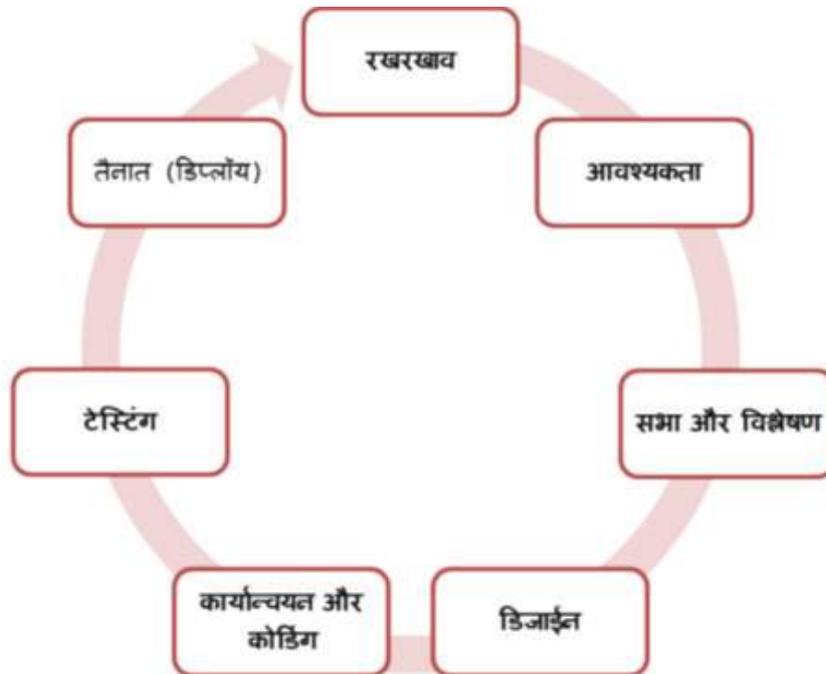
1. सॉफ्टवेयर डेवलपमेंट लाइफ साइकल को परिभाषित करें जिसमें बिजनेस रिक्वायरमेंट स्पेसिफिकेशन (बीआरएस), सॉफ्टवेयर रिक्वायरमेंट्स स्पेसिफिकेशन (एसआरएस), हाई-लेवल डिजाइन (एचएलडी), और लो-लेवल डिजाइन (एलएलडी) शामिल हैं।

### 7.1.1 सॉफ्टवेयर डेवलपमेंट लाइफ साइकल (चक्र) (एसडीएलसी)

सॉफ्टवेयर डेवलपमेंट लाइफसाइकल (एसडीएलसी) सॉफ्टवेयर डेवलपमेंट में शामिल चरणों को परिभाषित करने के लिए एक रूपरेखा है। यह सॉफ्टवेयर को विकसित करने, तैनात करने और बनाए रखने के लिए विस्तृत योजना का वर्णन करता है। एसडीएलसी पूरे डेवलपमेंट चक्र को परिभाषित करता है, जिसमें उत्पाद रखरखाव के लिए आवश्यकताओं को इकट्ठा करने में शामिल सभी कार्य शामिल हैं। एसडीएलसी का लक्ष्य एक उच्च गुणवत्ता वाले उत्पाद को वितरित करना है जो ग्राहक की जरूरतों को पूरा करता है। SDLC के चरण इस प्रकार हैं: आवश्यकता एकत्र करने, डिजाइन, कोडिंग, परीक्षण और रखरखाव। उत्पाद को व्यवस्थित रूप से प्रदान करने के लिए चरणों का पालन करना महत्वपूर्ण है। उदाहरण के लिए, सॉफ्टवेयर विकसित किया जाना चाहिए, और ऐसा करने के लिए स्वतंत्र होने के दौरान उत्पाद की एक विशिष्ट विशेषता पर काम करने के लिए एक टीम बनाई जाती है।

### 7.1.2 एसडीएलसी चक्र

SDLC चक्र सॉफ्टवेयर डेवलपमेंट प्रक्रिया का प्रतिनिधित्व करता है और नीचे आरेखीय रूप से चित्रित किया गया है:



चित्र 7.1 SDLC चक्र

### 7.1.3 व्यावसायिक आवश्यकताएँ विशिष्टता (बीआरएस)

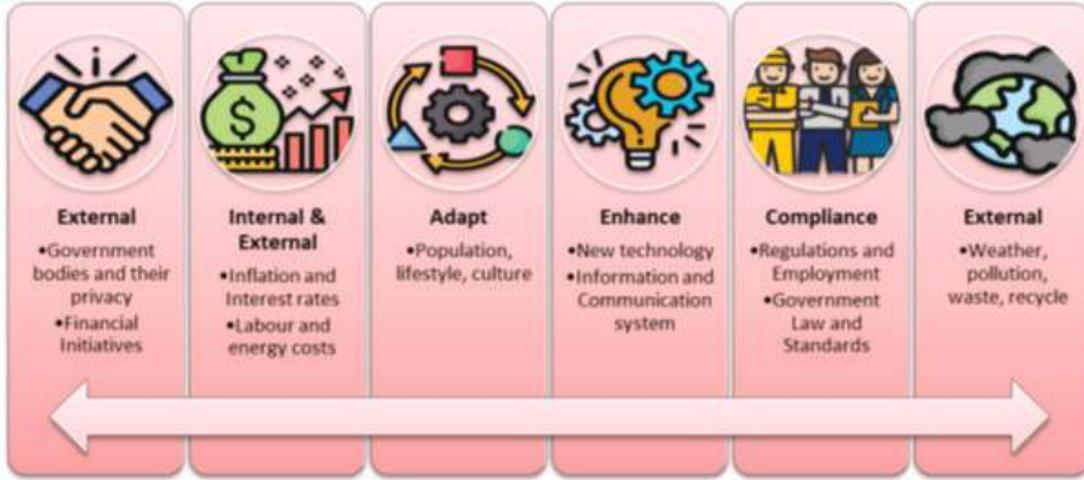
ये ऐसे दस्तावेज हैं जो उन व्यावसायिक उद्देश्यों को सूचीबद्ध करते हैं जिन्हें ग्राहक प्राप्त करने का प्रयास कर रहा है, साथ ही उत्पादों या सिस्टम के प्रमुख लक्ष्य और प्रदर्शन अपेक्षाएँ भी। यह यह भी दर्शाता है कि व्यापक स्तर पर ग्राहक की व्यावसायिक आवश्यकताओं को कैसे पूरा किया जाता है। इसे एक उच्च-स्तरीय दस्तावेज़ के रूप में जाना जाता है और यह सबसे व्यापक रूप से उपयोग किए जाने वाले और सार्वभौमिक रूप से स्वीकृत विनिर्देश दस्तावेज़ों में से एक है। आमतौर पर, व्यावसायिक आवश्यकता विशिष्टता (बीआरएस) परियोजना जीवन चक्र की शुरुआत में बनाई जाती है। एक व्यापार विश्लेषक का बीआरएस दस्तावेज़ आम तौर पर अन्य महत्वपूर्ण हितधारकों के विनिर्देशों पर आधारित होता है जो ग्राहक कंपनी के संपूर्ण विश्लेषण के बाद अंतिम उत्पाद के साथ बातचीत करेंगे। क्लाइंट तब दस्तावेज़ के अंतिम मसौदे पर जाता है ताकि यह सुनिश्चित हो सके कि यह व्यावसायिक हितधारकों की सभी अपेक्षाओं को पूरा करता है।

उत्पाद निवेशक, मध्य और उच्च प्रबंधन, और व्यापार विश्लेषक बीआरएस दस्तावेज़ों के प्राथमिक उपयोगकर्ता हैं। एक व्यावसायिक आवश्यकता विशिष्टता (बीआरएस) दस्तावेज़ में आम तौर पर परियोजना का पूरा दायरा, प्रदर्शन आवश्यकताओं और प्रयोज्यता, उत्पाद का उद्देश्य, उत्पाद के कार्य, उत्पाद की विशेषताएँ और कोई अन्य प्रासंगिक जानकारी शामिल होती है।



चित्र 7.2 व्यावसायिक आवश्यकताएँ विशिष्टता की विशेषताएँ

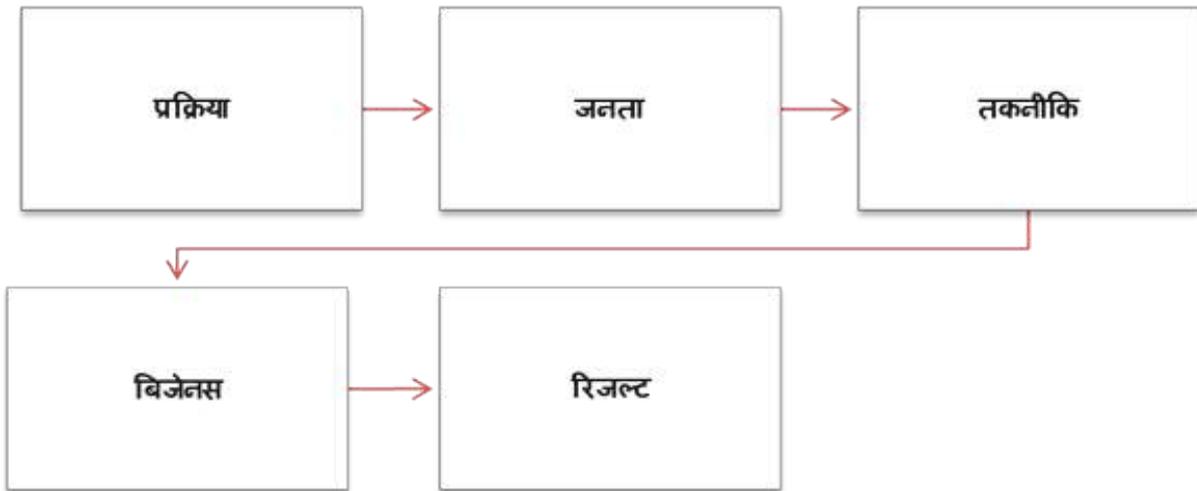
व्यापार की आवश्यकताएँ



चित्र 7.3 व्यापार की आवश्यकताएँ

7.1.4 व्यावसायिक आवश्यकताएँ प्रक्रिया

यह किसी विशेष व्यावसायिक लक्ष्य के लिए आवश्यकताओं की पहचान, विश्लेषण, परिभाषित और दस्तावेजीकरण की प्रक्रिया है। और यह परियोजना के दायरे को स्पष्ट और सटीक रूप से परिभाषित करने की प्रक्रिया है ताकि इसे पूरा करने के लिए आवश्यक समय-सीमा और संसाधनों का आकलन किया जा सके। CONOPS के समान, व्यावसायिक आवश्यकताएँ, जिन्हें हितधारक आवश्यकताएँ विनिर्देश (StRS) के रूप में भी जाना जाता है, सिस्टम के अंतिम उपयोगकर्ता के दृष्टिकोण से प्रस्तावित प्रणाली की विशेषताओं का वर्णन करती हैं। व्यावसायिक आवश्यकताओं को वितरित करने, संतुष्ट करने या पूरा करने के तरीकों में उत्पाद, सिस्टम, सॉफ्टवेयर और प्रक्रियाएं शामिल हैं।



चित्र 7.4 व्यावसायिक आवश्यकताएँ प्रक्रिया

### 7.1.5 सॉफ्टवेयर आवश्यकता विनिर्देश

सॉफ्टवेयर आवश्यकताएँ विनिर्देश (एसआरएस) एक दस्तावेज है जो किसी उत्पाद की मुख्य कार्यक्षमता और व्यावसायिक उद्देश्य का वर्णन करता है, सॉफ्टवेयर जिसे विकसित किया जाना है, और सॉफ्टवेयर के मुख्य कार्यों को कैसे किया जाना है। यह एक परियोजना का सारांश है जिसमें प्रौद्योगिकी की विशेषताएं और साथ ही वांछित व्यावसायिक लक्ष्य भी शामिल हैं। एसआरएस दस्तावेज़ महत्वपूर्ण है क्योंकि यह इस बात को पाटता है कि व्यवसाय को विकसित किए जा रहे सॉफ्टवेयर की विशेषताओं, व्यापक लेआउट और वर्कफ्लोज़ को दस्तावेज करके व्यापार को क्या मिलेगा और व्यवसाय क्या चाहता है। सॉफ्टवेयर आवश्यकताएँ विनिर्देश (SRS) किसी भी परियोजना के आधार के रूप में कार्य करता है क्योंकि इसमें एक ढांचा होता है जो प्रत्येक टीम के सदस्य का पालन करेंगे।

सिस्टम विश्लेषक संबंधित हितधारकों से सभी आवश्यक जानकारी एकत्र करने के प्रभारी हैं। वे फिर अन्य विभागों के साथ जानकारी साझा करते हैं। सॉफ्टवेयर आवश्यकता विनिर्देश के लिए प्राथमिक दर्शक परियोजना प्रबंधक (एसआरएस) हैं। SRS में कार्यात्मक और गैर-कार्यात्मक आवश्यकताएं शामिल हैं। SRS निराशा और गैर-उत्पादक समय को कम करके सॉफ्टवेयर विकास में एक महत्वपूर्ण लाभ प्रदान करता है। यह टीम के सदस्यों को एक साथ काम करने में भी सहायता करता है ताकि यह सुनिश्चित किया जा सके कि सभी आवश्यकताएं पूरी हो। उत्पाद आवश्यकताएँ दस्तावेज़ (PRD) सॉफ्टवेयर आवश्यकता विनिर्देश (SRS) (PRD) के लिए एक और नाम है। एक प्रस्तावित सॉफ्टवेयर जो किसी कर्मचारी के कार्यालय समय को ट्रैक करने के लिए डिज़ाइन किया गया है, एक सॉफ्टवेयर आवश्यकता विनिर्देश (एसआरएस) दस्तावेज़ का एक उदाहरण है। इस मामले में, दस्तावेज़ में लॉगिन मॉड्यूल, कर्मचारी मॉड्यूल, व्यवस्थापक मॉड्यूल और रिपोर्टिंग मॉड्यूल शामिल होना चाहिए।

यह सॉफ्टवेयर में मौजूद कार्यात्मक और गैर-कार्यात्मक आवश्यकताओं को विनिर्देश करता है।	यह विज्ञान-रिचार्जमेंटस स्पेसिफिकेशन (पीआरएस) से लिया गया है।	इसे सिस्टम एनालिस्ट या सिस्टम अर्किटेक्ट द्वारा बनाया जाता है। हालांकि, कुछ कंपनियों में, व्यापार विश्लेषक एक SRS भी बना सकते हैं।
यह उच्च स्तर पर सॉफ्टवेयर के तकनीकी और कार्यात्मक विनिर्देशों का भी वर्णन करता है।	यह इन संसाधनों से संबंधित है जो कंपनी प्रदान करती है।	यह मतलब है कि सॉफ्टवेयर या एप्लिकेशन का उपयोग करते समय व्यवसाय कैसे काम करता है।
देवास और उपयोग के मामले शामिल हैं।	यह कार्यात्मक आवश्यकताओं और गैर-कार्यात्मक आवश्यकताओं की व्याख्या करता है।	यह एक संपूर्ण दस्तावेज़ है जो बताता है कि सिस्टम विकसित होने पर कैसे व्यवहार करता है।
ताकिकों और आरेखों के संदर्भ शामिल हैं।	यह किसी भी उपयोगकर्ता अपार या क्लाइंट को सूचीबद्ध नहीं करता है।	SRS दस्तावेज़ व्यवसाय आवश्यकताओं विशिष्टता (BRS) से बना किया गया है।

चित्र 7.5 सॉफ्टवेयर आवश्यकता विशिष्टता की विशेषताएं

SRS में उपयुक्त चरण :



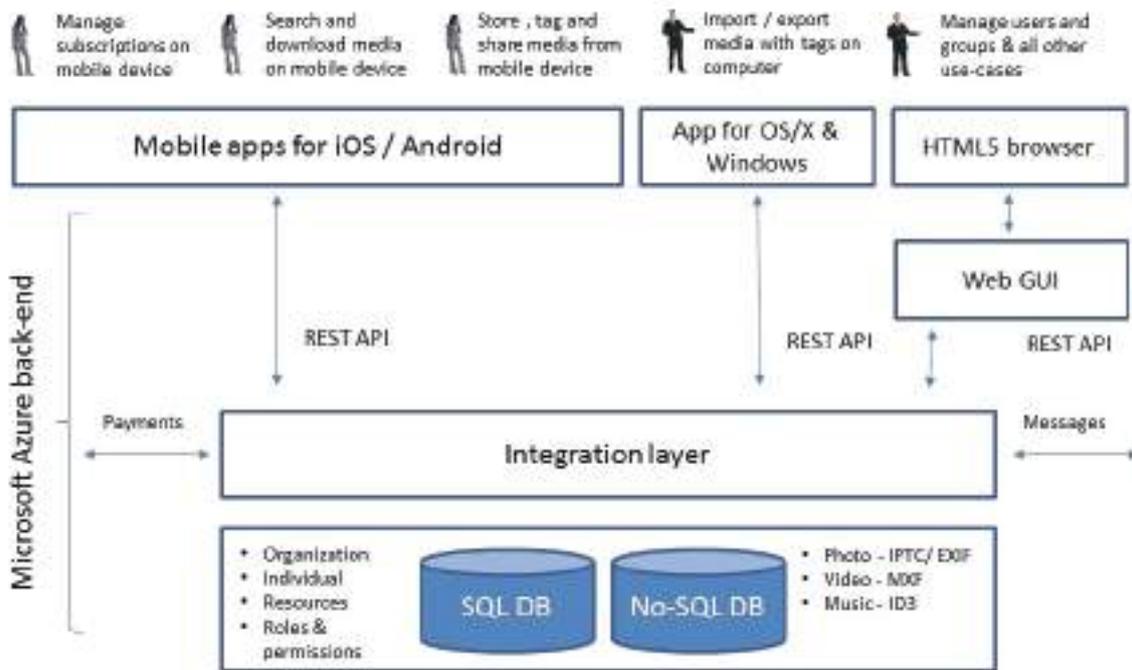
चित्र 7.6 SRS के चरण

### 7.1.6 उच्च-स्तरीय डिजाइन

HLD "सामान्य सिस्टम डिजाइन" के लिए एक संक्षिप्त नाम है, जो अनुप्रयोग के समग्र विवरण और वास्तुकला को संदर्भित करता है। यह सिस्टम आर्किटेक्चर, डेटाबेस डिजाइन और सिस्टम, सेवाओं, प्लेटफार्मों और मॉड्यूल संबंधों के सारांश का वर्णन करता है। HLD, या उच्च-स्तरीय डिजाइन, एक सामान्य डिजाइन प्रणाली है। इसे सिस्टम या मैक्रो-लेवल डिजाइन के रूप में भी जाना जाता है। एक उच्च-स्तरीय डिजाइन में सिस्टम आर्किटेक्चर का एक समग्र विवरण, इसके डेटाबेस का डिजाइन और इसकी सेवाओं, सिस्टम, उपयोग किए जाने वाले प्लेटफार्मों और मॉड्यूल संबंधों का संक्षिप्त विवरण शामिल है। एक समाधान वास्तुकार HLD बनाता है। यह अनिवार्य रूप से समग्र ग्राहक/व्यावसायिक आवश्यकताओं को उच्च-स्तरीय समाधान में अनुवाद करता है। यह निम्न-स्तरीय डिजाइन से पहले चित्र में प्रवेश करता है।

एचएलडी के रूप में इसका एक सामान्य संक्षिप्त नाम है।	यह डिजाइन की एक सामान्य प्रणाली है जो संपूर्ण प्रणाली से संबंधित है।	एचएलडी समग्र रूप से एक प्रणाली/अनुप्रयोग के समग्र आर्किटेक्चर/विवरण का वर्णन करता है।	यह संक्षेप में व्यक्तिगत मॉड्यूल की कार्यक्षमता का वर्णन करता है।
समाधान आर्किटेक्ट एचएलडी बनाते हैं।	एक उच्च-स्तरीय डिजाइन में भाग लेने वाले क्लाइंट टीम, समीक्षा टीम और डिजाइन टीम हैं।	एलएलडी से पहले एचएलडी बनाने और लागू करने की जरूरत है।	जेएलडी के मामले में इनपुट मानदंड एसआरएस (सॉफ्टवेयर रिक्वायरमेंट स्पेसिफिकेशन) है।
	यह क्लाइंट/व्यावसायिक आवश्यकता को उच्च स्तरीय समाधान में परिवर्तित करता है।	एचएलडी के आउटपुट मानदंड समीक्षा रिकॉर्ड, फंक्शन डिजाइन और डेटाबेस डिजाइन हैं।	

चित्र 7.7 उच्च-स्तरीय डिजाइन की विशेषताएं



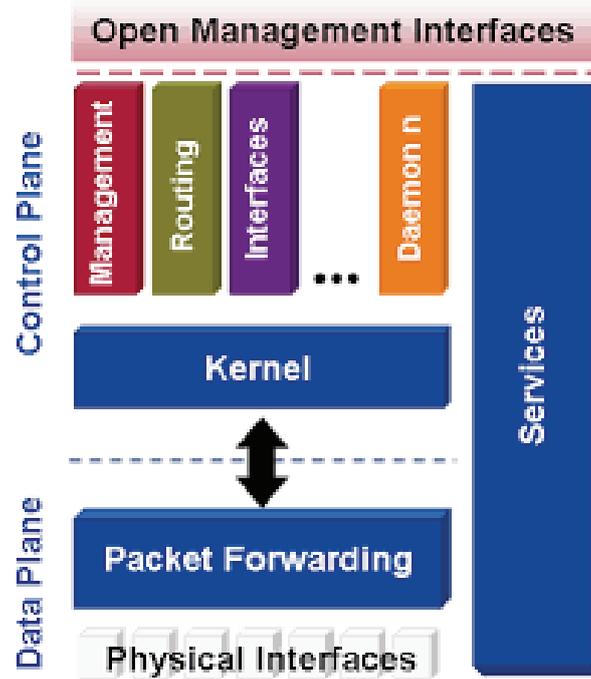
चित्र 7.8 उच्च स्तरीय डिजाइन का उदाहरण (विघटनकारी वास्तुकला)  
(Image Source: disruptivearchitecture.info)

### 7.1.7 निम्न-स्तरीय डिजाइन

LLD, या निम्न-स्तरीय डिजाइन, अनिवार्य रूप से प्रत्येक मॉड्यूल का एक विस्तृत विवरण है। दूसरे शब्दों में, एलएलडी प्रत्येक मॉड्यूल का पूरी तरह से वर्णन करता है, जिसमें प्रत्येक सिस्टम घटक का वास्तविक तर्क शामिल है। यह प्रत्येक सिस्टम विनिर्देश के साथ बहुत विस्तार से जाता है, इसे मोनिकर विस्तृत/सूक्ष्म स्तर के डिजाइन को अर्जित करता है। निम्न-स्तरीय डिजाइन डिजाइनरों और डेवलपर्स द्वारा बनाया और कार्यान्वित किया जाता है। यह एक उच्च-स्तरीय समाधान को एक बहुत विस्तृत समाधान में बदल सकता है। नतीजतन, LLD HLD के डिजाइन और कार्यान्वित होने के बाद ही खेल में आता है।



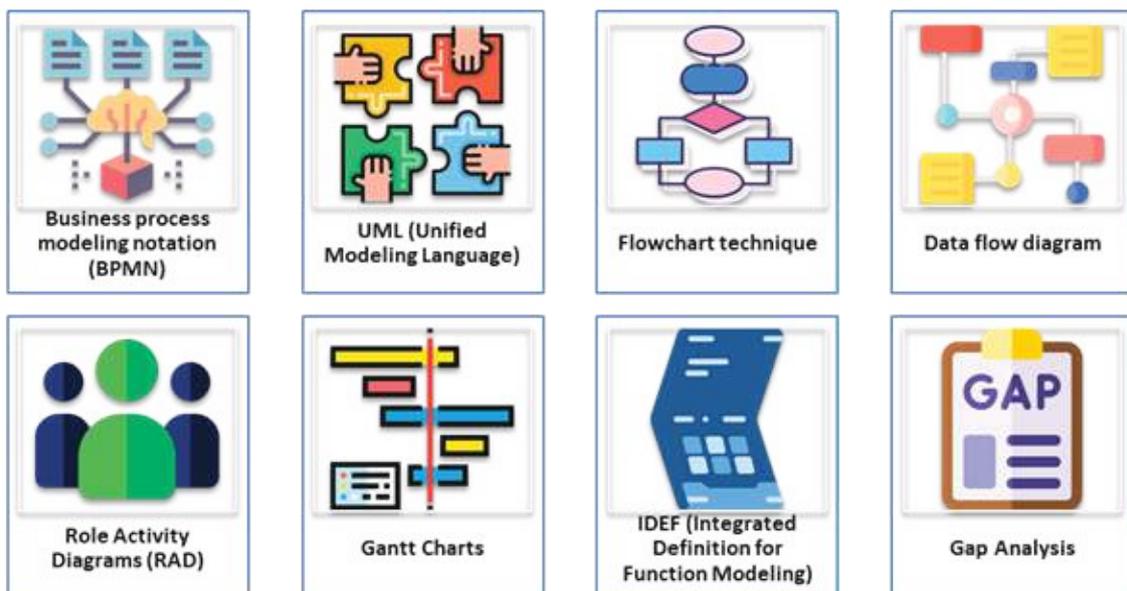
चित्र 7.9 निम्न स्तर के डिजाइन की मुख्य विशेषताएं



चित्र 7.10 निम्न स्तरीय डिज़ाइन दस्तावेज़ नमूना

(Image Source: enterstarcrypticity.blogspot.com)

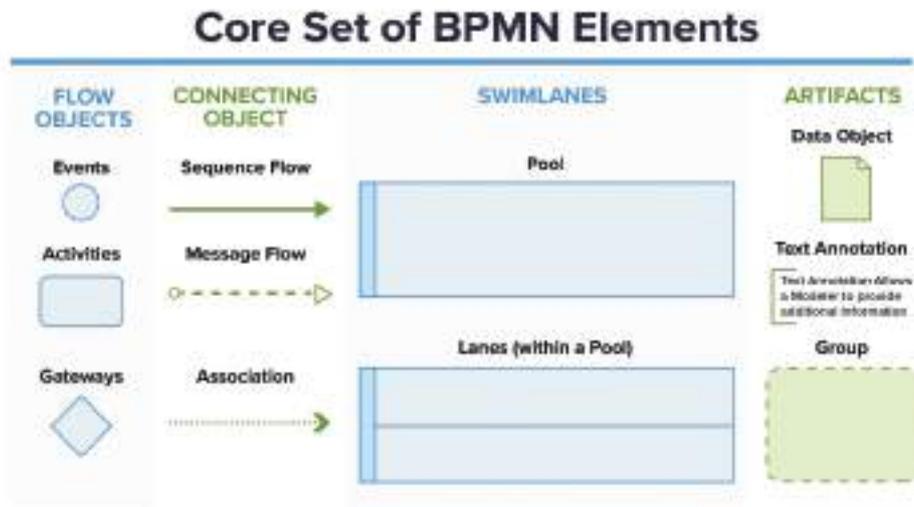
**7.1.8 आवश्यकता विश्लेषण के लिए प्रयुक्त विभिन्न तकनीकें आवश्यकता विश्लेषण के लिए विभिन्न तकनीकों का उपयोग किया जाता है। नीचे विभिन्न आवश्यकता विश्लेषण तकनीकों की सूची दी गई है**



चित्र 7.11 आवश्यकता विश्लेषण के लिए प्रयुक्त तकनीकें

### 1. व्यावसायिक प्रक्रिया मॉडलिंग अंकन (BPMN)

यह विधि प्रक्रिया फ्लोचार्ट बनाने के समान है, हालांकि BPMN के अपने प्रतीक और तत्व हैं। व्यावसायिक प्रक्रियाओं के लिए ग्राफ व्यावसायिक प्रक्रिया मॉडलिंग और संकेतन का उपयोग करके बनाया जाता है। ये रेखांकन व्यावसायिक प्रक्रिया को समझना आसान बनाते हैं। BPMN एक प्रसिद्ध प्रक्रिया सुधार पद्धति है।

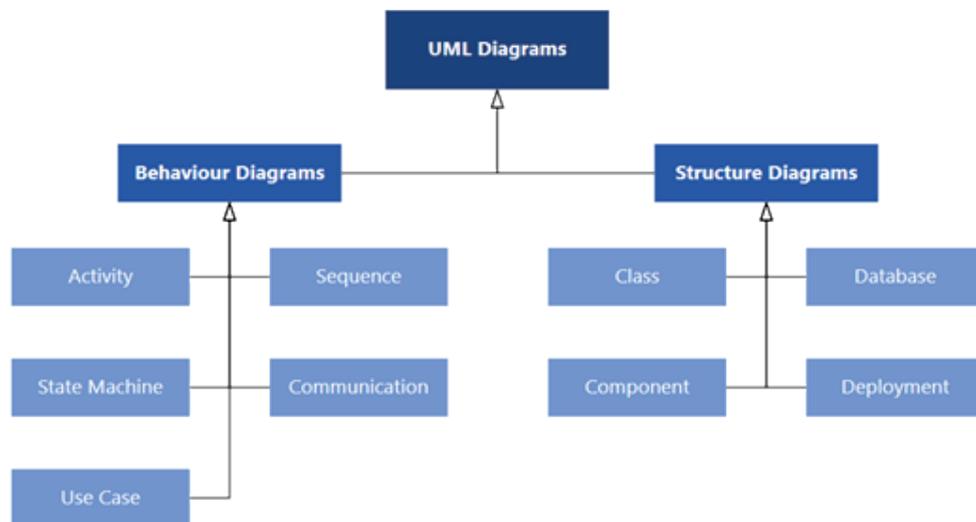


चित्र 7.12 व्यावसायिक प्रक्रिया मॉडलिंग अंकन (BPMN)

(Image Source: processmaker.com)

### 2. यूएमएल (एकीकृत मॉडलिंग भाषा)

UML एक सॉफ्टवेयर सिस्टम की कलाकृतियों को निर्दिष्ट, कल्पना, निर्माण और दस्तावेज़ करने के लिए उपयोग किए जाने वाले आरेखों का एक एकीकृत सेट है। ऑब्जेक्ट-ओरिएंटेड सॉफ्टवेयर विकसित करते समय और सॉफ्टवेयर डेवलपमेंट प्रक्रिया के साथ काम करते समय, यूएमएल एक उपयोगी तकनीक है। UML एक सॉफ्टवेयर प्रोजेक्ट के डिजाइन का प्रतिनिधित्व करने के लिए ग्राफिकल नोटेशन को नियुक्त करता है। UML भी सॉफ्टवेयर के वास्तुशिल्प डिजाइन के सत्यापन में सहायता करता है।

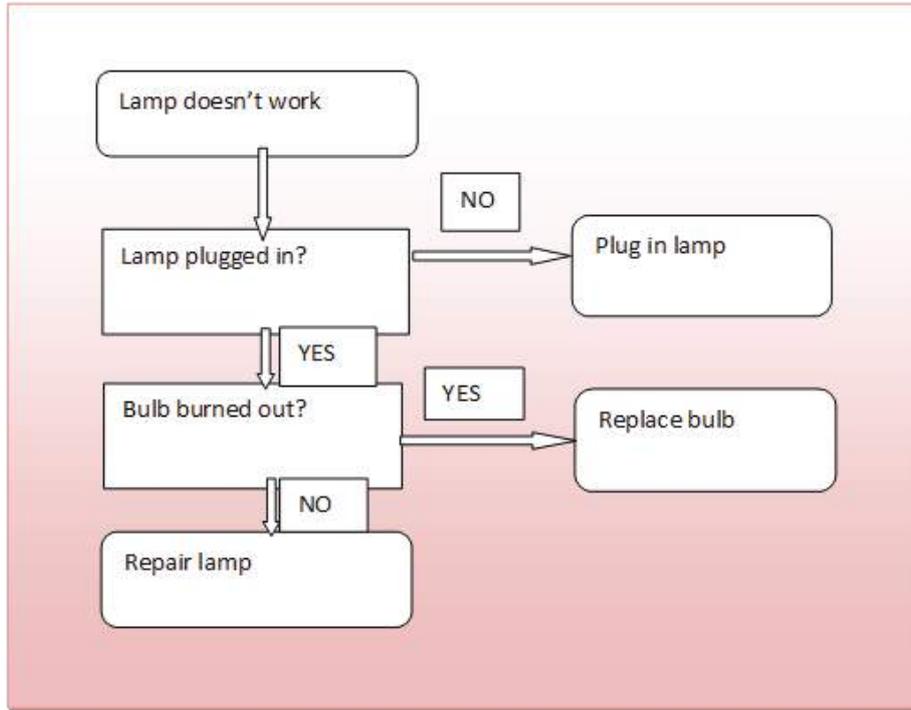


चित्र 7.13 यूएमएल (एकीकृत मॉडलिंग भाषा)

(image Source: Microsoft Support)

### 3. फ्लोचार्ट तकनीक

एक फ्लोचार्ट संबंधित गतिविधियों के समूह के अनुक्रमिक प्रवाह और नियंत्रण तर्क को दर्शाता है और रैखिक, क्रॉस-फ़ंक्शनल और टॉप-डाउन सहित विभिन्न स्वरूपों में आता है। फ्लोचार्ट सिस्टम इंटरैक्शन, डेटा फ्लो इत्यादि को चित्रित कर सकता है। फ्लो चार्ट को समझना आसान है और तकनीकी और गैर-तकनीकी दोनों टीमों के सदस्यों द्वारा इसका उपयोग किया जा सकता है। फ्लोचार्ट तकनीक किसी प्रक्रिया की महत्वपूर्ण विशेषताओं को उजागर करने के लिए उपयोगी है।

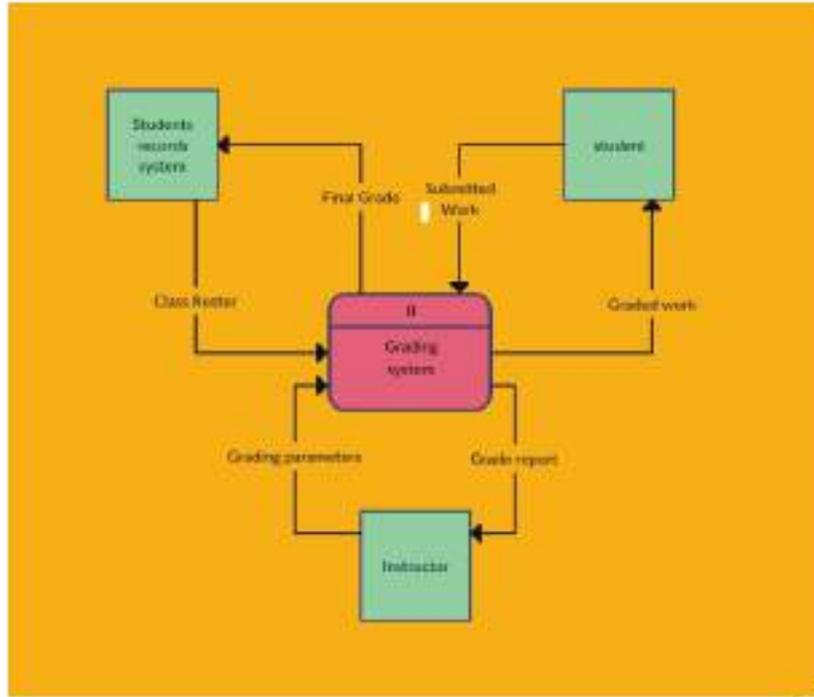


चित्र 7.14 फ्लोचार्ट तकनीक का उदाहरण

(Image Source: conceptDraw)

### 4. डेटा फ्लो आरेख

इस तकनीक का उपयोग जटिल और कठिन-से-वर्णन प्रणालियों और प्रक्रियाओं को नेत्रहीन रूप से प्रस्तुत करने के लिए किया जाता है। डेटा प्रवाह आरेख एक प्रक्रिया या प्रणाली के माध्यम से डेटा के प्रवाह को दर्शाते हैं। इसमें डेटा इनपुट और आउटपुट, डेटा स्टोर और विभिन्न उप प्रक्रियाएं भी शामिल हैं जिनसे डेटा गुजरता है। DFD विभिन्न संस्थाओं और उनके संबंधों का वर्णन करने के लिए मानकीकृत नोटेशन और प्रतीकों का उपयोग करता है। सिस्टम में खामियों की पहचान करना आसान होता है जब इसके सभी घटकों की कल्पना की जाती है। फिर एक मजबूत समाधान बनाने के लिए इन खामियों को दूर किया जाता है।

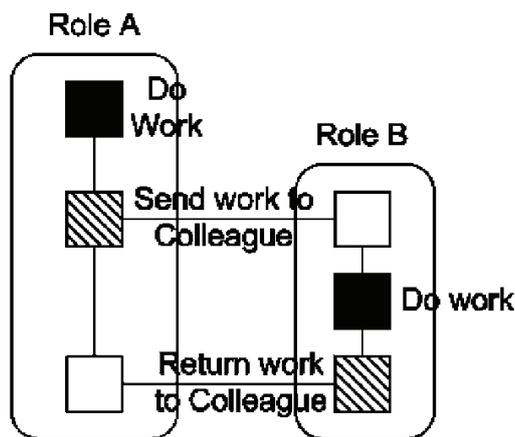


चित्र 7.15 डेटा फ्लो आरेख

(Image Source: creately.com)

### 5. भूमिका गतिविधि आरेख (आरएडी)

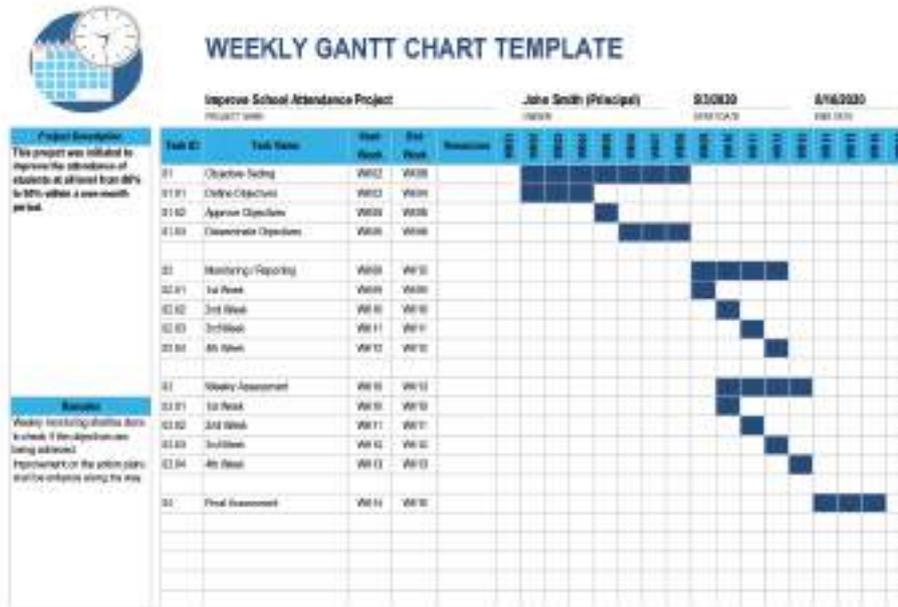
एक भूमिका-गतिविधि आरेख (आरएडी) एक नियम-उन्मुख प्रक्रिया मॉडल है जो भूमिका-गतिविधि आरेखों का प्रतिनिधित्व करता है। भूमिका गतिविधि आरेख एक संगठन की गतिशीलता और भूमिका संरचना का एक उच्च-स्तरीय दृश्य है। भूमिकाओं का उपयोग गतिविधियों को जिम्मेदारी की असतत इकाइयों में व्यवस्थित करने के लिए किया जाता है। गतिविधियाँ एक भूमिका के मूलभूत घटक हैं। एक गतिविधि अलगाव में या भूमिका के भीतर अन्य गतिविधियों के संयोजन में की जा सकती है।



चित्र 7.16 भूमिका गतिविधि आरेख (RAD)

### 6. गैन्ट चार्ट

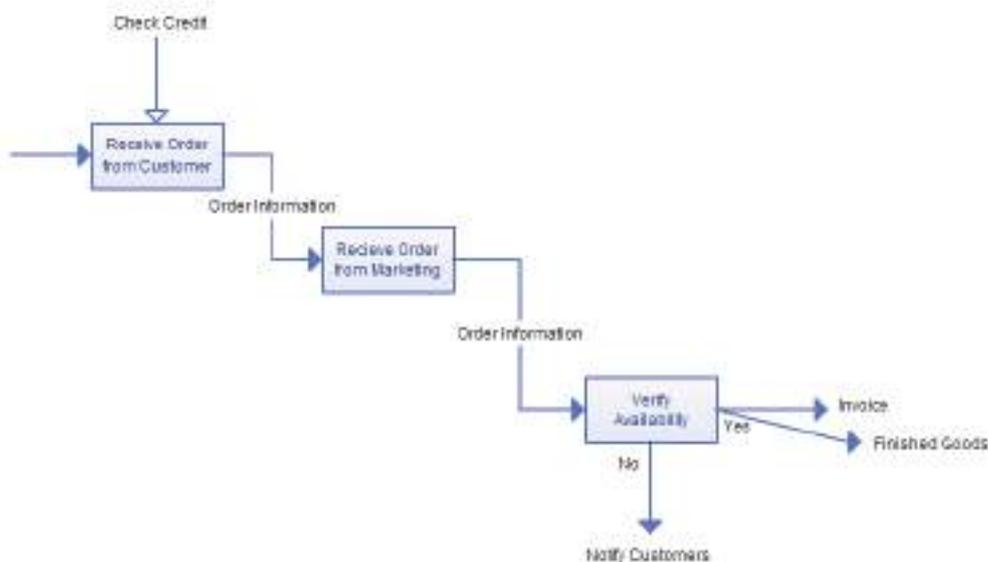
प्रोजेक्ट प्लानिंग में गैन्ट चार्ट का उपयोग किया जाता है क्योंकि वे अनुसूचित कार्यों के साथ -साथ समयसीमा का एक दृश्य प्रतिनिधित्व प्रदान करते हैं। गैन्ट चार्ट यह निर्धारित करने में सहायता करते हैं कि किस तिथि तक पूरा किया जाना है। प्रोजेक्ट के सभी कार्यों की शुरुआत और अंतिम तिथियां एक ही दृश्य में दिखाई देती हैं।



चित्र 7.17 गैन्ट चार्ट

### 7. आईडीईएफ (फंक्शन मॉडलिंग के लिए एकीकृत परिभाषा)

फंक्शन मॉडलिंग (IDEFM) तकनीक के लिए एकीकृत परिभाषा एक बॉक्स की मदद से एक प्रक्रिया के कार्यों और बच्चे और माता-पिता के सिस्टम के साथ उनके संबंधों का प्रतिनिधित्व करती है। यह एक संगठन की प्रणाली की समझ हासिल करने के लिए एक खाका प्रदान करता है।



चित्र 7.18 IDEF

(Image Link: [japaneseclass.jp](http://japaneseclass.jp))

### 8. गैप विश्लेषण

गैप विश्लेषण एक ऐसी तकनीक है जो सॉफ्टवेयर एप्लिकेशन के प्रदर्शन में अंतराल का विश्लेषण करने में मदद करती है ताकि यह निर्धारित किया जा सके कि व्यावसायिक आवश्यकताओं को पूरा किया गया है या नहीं। इसमें वे कदम भी शामिल हैं जो यह सुनिश्चित करने के लिए उठाए जाने हैं कि सभी व्यावसायिक आवश्यकताओं को सफलतापूर्वक पूरा किया जाए। गैप वर्तमान राज्य और लक्ष्य राज्य के बीच के अंतर को दर्शाता है। गैप विश्लेषण को आवश्यकता विश्लेषण, आवश्यकता मूल्यांकन या आवश्यकता-अंतर विश्लेषण के रूप में भी जाना जाता है।

गैप विश्लेषण प्रक्रिया:



चित्र 7.19 गैप विश्लेषण प्रक्रिया

Target Architecture Baseline Architecture ↓	Video Conferencing Services	Enhanced Telephony Services	Mailing List Services	Eliminated Services ↓
Broadcast Services				Intentionally eliminated
Video Conferencing Services	Included			
Enhanced Telephony Services		Potential match		
Shared Screen Services				Unintentionally excluded - a gap in Target Architecture
New →		Gap: Enhanced services to be developed or produced	Gap: To be developed or produced	

© The Open Group

चित्र 7.20 गैप विश्लेषण

(Image Source: Template Lab)

## सारांश



- सॉफ्टवेयर डेवलपमेंट लाइफसाइकल (एसडीएलसी) एक फ्रेमवर्क है जो सॉफ्टवेयर के विकास में शामिल चरणों को परिभाषित करता है। यह सॉफ्टवेयर के निर्माण, तैनाती और बनाए रखने के लिए विस्तृत योजना को कवर करता है।
- एसडीएलसी विकास के पूर्ण चक्र को परिभाषित करता है यानी किसी उत्पाद के रखरखाव के लिए आवश्यकता एकत्र करने में शामिल सभी कार्य।
- बिजनेस एनालिस्ट और प्रोजेक्ट मैनेजर ने ग्राहक के साथ एक बैठक स्थापित की, जो ग्राहक क्या बनाना चाहता है, जो कि एंड-यूजर होगा, और उत्पाद का उद्देश्य क्या है। किसी उत्पाद के निर्माण से पहले उत्पाद की एक मुख्य समझ या ज्ञान बहुत महत्वपूर्ण है।
- एसआरएस दस्तावेज़ में एकत्र की गई आवश्यकता का उपयोग इनपुट के रूप में किया जाता है, और सिस्टम विकास को लागू करने के लिए उपयोग किए जाने वाले सॉफ्टवेयर आर्किटेक्चर को प्राप्त किया जाता है।
- यूएटी के मामले में, उत्पादन वातावरण की एक प्रतिकृति बनाई जाती है और डेवलपर्स के साथ ग्राहक परीक्षण करता है। यदि ग्राहक अपेक्षित रूप से एप्लिकेशन को ढूँढता है, तो ग्राहक द्वारा लाइव जाने के लिए साइन-ऑफ प्रदान किया जाता है।
- कोडिंग पूरा होने के बाद परीक्षण शुरू हो जाता है और मॉड्यूल परीक्षण के लिए जारी किए जाते हैं। इस चरण में, विकसित सॉफ्टवेयर का अच्छी तरह से परीक्षण किया जाता है और पाए जाने वाले किसी भी दोष को डेवलपर्स को तय करने के लिए सौंपा जाता है।
- निम्न-स्तरीय डिजाइन, एलएलडी के रूप में संक्षिप्त, अनिवार्य रूप से प्रत्येक मॉड्यूल का एक विस्तृत विवरण है। दूसरे शब्दों में, LLD प्रत्येक मॉड्यूल को प्रत्येक सिस्टम घटक के वास्तविक तर्क को शामिल करके विस्तार से बताता है।
- उच्च-स्तरीय डिजाइन (HLD) सामान्य सिस्टम डिजाइन के लिए एक संक्षिप्त नाम है, जो समग्र सिस्टम डिजाइन को संदर्भित करता है। यह अनुप्रयोग के समग्र विवरण/वास्तुकला का वर्णन करता है। यह सिस्टम आर्किटेक्चर, डेटाबेस डिजाइन और सिस्टम, सेवाओं, प्लेटफार्मों के संक्षिप्त विवरण का वर्णन करता है।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

निम्न-स्तरीय डिजाइन और उच्च-स्तरीय डिजाइन के बीच अंतर।

सॉफ्टवेयर आवश्यकताओं के विनिर्देशों का वर्णन करें।

बीआरएस में व्यावसायिक आवश्यकताएं क्या हैं?

रिक्त स्थान भरें:

\_\_\_\_\_ एक तकनीक जो व्यावसायिक आवश्यकताओं को पूरा करने या नहीं होने के लिए यह निर्धारित करने के लिए एक सॉफ्टवेयर एप्लिकेशन के प्रदर्शन में अंतराल का विश्लेषण करने में मदद करता है।

\_\_\_\_\_ तकनीकी एक बॉक्स की मदद से बच्चे और मूल प्रणालियों के लिए एक प्रक्रिया और उनके संबंधों के कार्यों का प्रतिनिधित्व करता है।

फ्लोचार्ट विभिन्न स्वरूपों जैसे कि \_\_\_\_\_, \_\_\_\_\_, और \_\_\_\_\_ में हैं।

MCQ प्रश्न :

1. सॉफ्टवेयर विकास जीवनचक्र में पहला कदम क्या है?

प्रणाली की रूपरेखा

कोडन

सिस्टम परीक्षण

प्रारंभिक जांच और विश्लेषण

2. निम्नलिखित में से कौन सॉफ्टवेयर डेवलपमेंट लाइफ साइकिल (SDLC) के सिस्टम प्लानिंग और डिजाइनिंग चरण में शामिल है?

आकार

समानांतर रन

विशिष्टता फ्रीज

ऊपर के सभी

3. निम्नलिखित में से कौन सा प्रोटोटाइप प्रोटोटाइप मॉडल से जुड़ा नहीं है?

डोमेन प्रोटोटाइप

लंबवत प्रोटोटाइप

क्षैतिज प्रोटोटाइप

विकर्ण प्रोटोटाइप

## यूनिट 7.2: सॉफ्टवेयर डेवलपमेंट प्रक्रिया को मापने के लिए एलिमेंट

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. सॉफ्टवेयर विकास प्रक्रिया के विभिन्न पहलुओं को मापने के लिए तत्वों को वर्गीकृत करने में

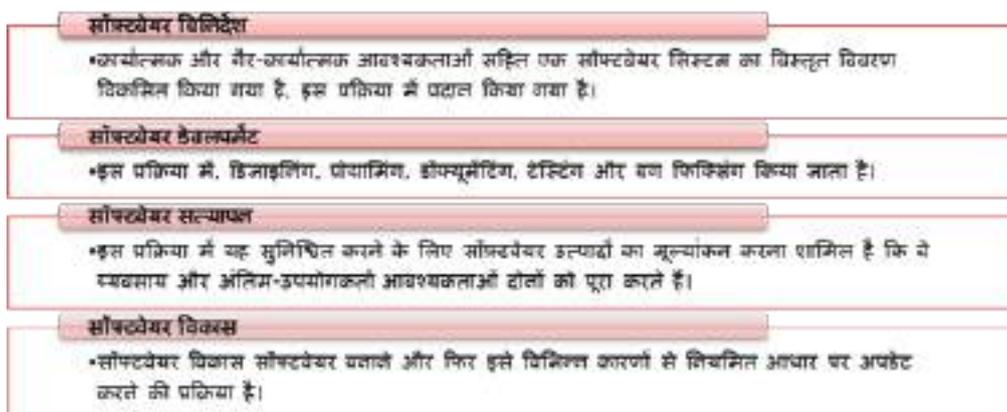
### 7.2.1 सॉफ्टवेयर डेवलपमेंट प्रोसेस (प्रक्रिया)

कंप्यूटर सिस्टम को नियंत्रित करने वाले कार्यक्रमों के रूप में निर्देशों का सेट और हार्डवेयर घटकों को संसाधित करने को सॉफ्टवेयर के रूप में संदर्भित किया जाता है। सॉफ्टवेयर उत्पाद बनाने के लिए गतिविधियों के एक सेट का उपयोग किया जाता है। इस संग्रह को एक सॉफ्टवेयर प्रक्रिया के रूप में जाना जाता है। डिजाइनिंग, प्रोग्रामिंग, डॉक्यूमेंटिंग, टेस्टिंग और बग फिक्सिंग सभी सॉफ्टवेयर डेवलपमेंट प्रक्रिया का हिस्सा हैं। सॉफ्टवेयर घटक: तीन सॉफ्टवेयर घटक हैं: उनमें से तीन हैं: कार्यक्रम, प्रलेखन और संचालन प्रक्रियाएं।



चित्र 7.21 सॉफ्टवेयर के घटक

निम्नलिखित आंकड़ा सॉफ्टवेयर विकास प्रक्रिया में चार बुनियादी प्रमुख प्रक्रिया गतिविधियों की व्याख्या करता है।



चित्र 7.22 प्रमुख प्रक्रिया गतिविधियाँ

निम्नलिखित आंकड़ा सॉफ्टवेयर संकट की व्याख्या करता है:



चित्र 7.23 सॉफ्टवेयर संकट

## 7.2.2 सॉफ्टवेयर मापन मेट्रिक्स

एक सॉफ्टवेयर मीट्रिक सॉफ्टवेयर विशेषताओं का एक औसत दर्जे का या गिनती योग्य उपाय है। सॉफ्टवेयर मेट्रिक्स विभिन्न प्रकार के उद्देश्यों के लिए उपयोगी हैं, जिनमें सॉफ्टवेयर प्रदर्शन को मापना, काम की वस्तुओं की योजना बनाना, उत्पादकता की गणना करना, और कई अन्य शामिल हैं। कई मेट्रिक्स सॉफ्टवेयर विकास प्रक्रिया से जुड़े हैं। सॉफ्टवेयर मेट्रिक्स चार प्रबंधन कार्यों के अनुरूप हैं: योजना, संगठन, नियंत्रण और सुधार।

सॉफ्टवेयर मेट्रिक्स को दो श्रेणियों में विभाजित किया गया है:

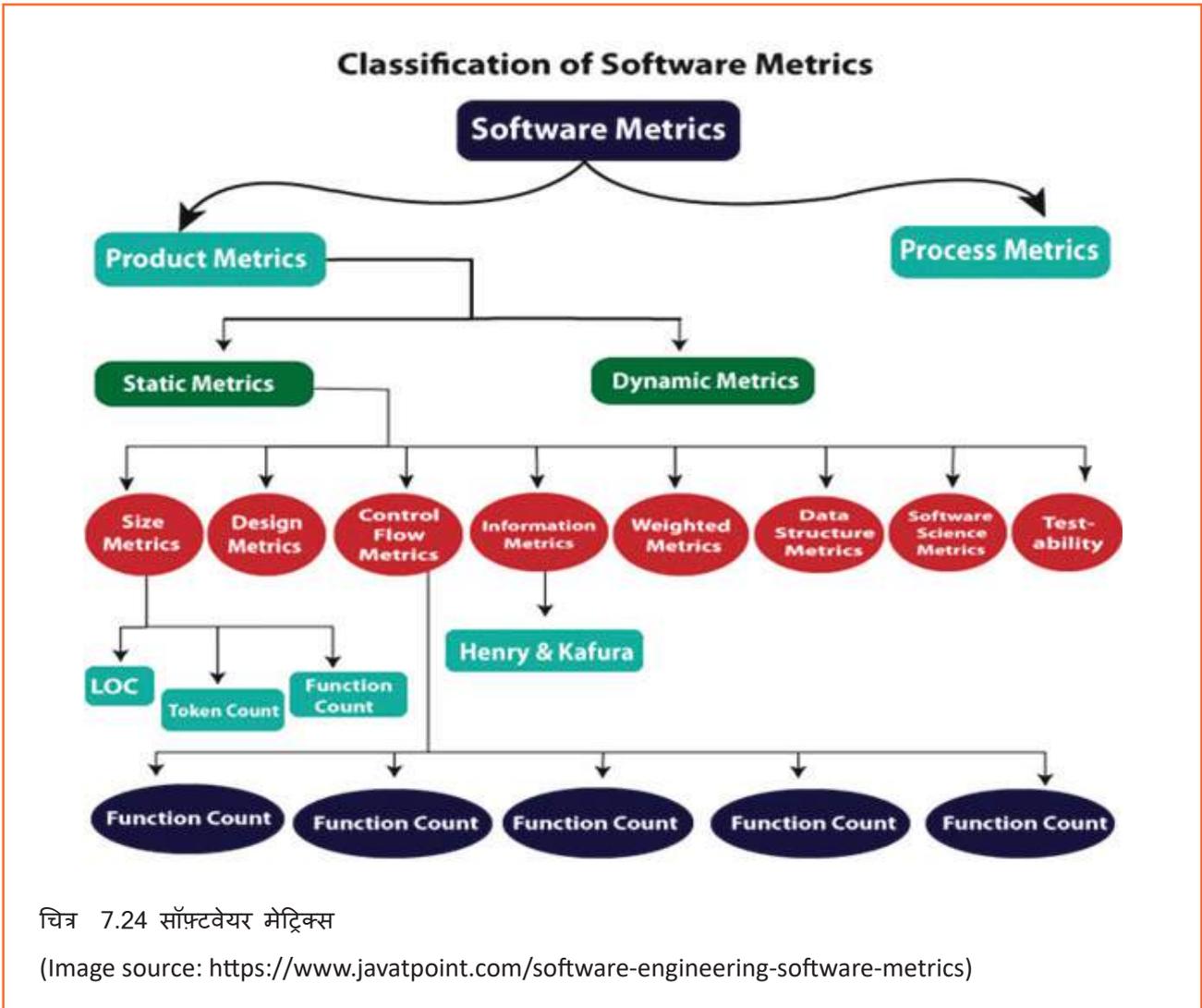
**उत्पाद मेट्रिक्स:** ये सॉफ्टवेयर उत्पाद के विभिन्न पहलुओं के माप हैं। निम्नलिखित दो सबसे महत्वपूर्ण सॉफ्टवेयर विशेषताएं हैं:

आकार और जटिलता

गुणवत्ता और विश्वसनीयता

इन मेट्रिक्स की गणना SDLC के विभिन्न चरणों में की जा सकती है।

**प्रक्रिया मेट्रिक्स:** ये सॉफ्टवेयर विकास प्रक्रिया के विभिन्न पहलुओं के माप हैं। गलती का पता लगाने की प्रभावशीलता पर विचार करें। उनका उपयोग सॉफ्टवेयर विकास विधियों, तकनीकों और उपकरणों की विशेषताओं का आकलन करने के लिए किया जाता है।



चित्र 7.24 सॉफ्टवेयर मेट्रिक्स

(Image source: <https://www.javatpoint.com/software-engineering-software-metrics>)

## सारांश

- प्रोग्राम के रूप में निर्देशों का वह समूह जो कंप्यूटर सिस्टम को नियंत्रित करता है और हार्डवेयर घटकों को प्रोसेस करता है, सॉफ्टवेयर कहलाता है। सॉफ्टवेयर उत्पाद बनाने के लिए गतिविधियों के एक सेट का उपयोग किया जाता है। इस संग्रह को एक सॉफ्टवेयर प्रक्रिया के रूप में जाना जाता है। डिजाइनिंग, प्रोग्रामिंग, दस्तावेजीकरण, परीक्षण और बग फिक्सिंग सभी सॉफ्टवेयर विकास प्रक्रिया का हिस्सा हैं।
- सॉफ्टवेयर घटक: तीन सॉफ्टवेयर घटक हैं: उनमें से तीन हैं: प्रोग्राम, दस्तावेजीकरण, और संचालन प्रक्रियाएं।
- कंप्यूटर प्रोग्राम निर्देशों की एक सूची है जो कंप्यूटर को बताता है कि क्या करना है।
- उत्पाद के बारे में स्रोत जानकारी डिजाइन दस्तावेजों, विस्तृत कोड टिप्पणियों आदि में निहित है।
- सॉफ्टवेयर विनिर्देश - इस प्रक्रिया में कार्यात्मक और गैर-कार्यात्मक आवश्यकताओं सहित विकसित किए जाने वाले सॉफ्टवेयर सिस्टम का विस्तृत विवरण प्रदान किया गया है।
- सॉफ्टवेयर डेवलपमेंट - इस प्रक्रिया में डिजाइनिंग, प्रोग्रामिंग, डॉक्यूमेंटिंग, टेस्टिंग और बग फिक्सिंग की जाती है।
- सॉफ्टवेयर सत्यापन - इस प्रक्रिया में यह सुनिश्चित करने के लिए सॉफ्टवेयर उत्पादों का मूल्यांकन करना शामिल है कि वे व्यवसाय और अंतिम-उपयोगकर्ता दोनों आवश्यकताओं को पूरा करते हैं।
- सॉफ्टवेयर इवोल्यूशन - सॉफ्टवेयर इवोल्यूशन सॉफ्टवेयर बनाने और फिर उसे विभिन्न कारणों से नियमित आधार पर अपडेट करने की प्रक्रिया है।
- एक सॉफ्टवेयर मीट्रिक सॉफ्टवेयर विशेषताओं का एक मापने योग्य या गणनीय माप है। सॉफ्टवेयर मेट्रिक्स विभिन्न उद्देश्यों के लिए उपयोगी होते हैं, जिसमें सॉफ्टवेयर प्रदर्शन को मापना, कार्य आइटम की योजना बनाना, उत्पादकता की गणना करना और कई अन्य शामिल हैं।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

1. वाटरफॉल मॉडल का वर्णन करें।
2. सॉफ्टवेयर प्रोसेस मॉडल क्या है?
3. सॉफ्टवेयर डेवलपमेंट प्रक्रिया क्या है?

रिक्त स्थान भरें :

\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, और \_\_\_\_\_ सॉफ्टवेयर विकास प्रक्रिया के सभी भाग हैं।

\_\_\_\_\_ सॉफ्टवेयर डेवलपमेंट की एक निरंतर प्रक्रिया, और यह विकास के साथ समवर्ती रूप से होता है।

किसी भी प्रक्रिया का पहला कदम हमेशा \_\_\_\_\_ होता है।

\_\_\_\_\_ प्रक्रिया एक पूरे के रूप में उत्पाद, परियोजना और डिजाइन को बेहतर बनाने के लिए सॉफ्टवेयर विकास को छोटे, अनुक्रमिक चरणों में विभाजित करता है।

\_\_\_\_\_ किसी भी व्यवसाय या व्यक्तिगत उद्देश्यों की जरूरतों को पूरा करने के लिए सॉफ्टवेयर प्रोग्राम या अनुप्रयोगों को विकसित करने की पुनरावृत्ति तार्किक प्रक्रिया के लिए संदर्भित है।





## 8. नौकरी की भूमिका के लिए कार्य आवश्यकताएँ, उपकरण और सॉफ्टवेयर



IT - ITeS SSC  
NASSCOM

यूनिट 8.1 कार्यक्रम विशिष्टता और प्रोग्रामिंग भाषा

यूनिट 8.2 अनुप्रयोगों और प्रमुख प्रक्रियाओं को विकसित करने के लिए दृष्टिकोण



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. पुनर्विकास प्रक्रिया के लिए DBMS और डेटा डिज़ाइन सहित डेटाबेस कौशल का निर्माण करने में
2. स्रोत कोडिंग मानकों, टिकटिंग टूल और अन्य आईटी-संबंधित तकनीकों के अनुप्रयोग का प्रदर्शन करने में

## यूनिट 8.1: कार्यक्रम विशिष्टता और प्रोग्रामिंग भाषा

### यूनिट के उद्देश्य



इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. विस्तृत प्रोग्राम विनिर्देश पढ़ने के तरीकों पर चर्चा करें और प्रोग्रामिंग भाषा का उपयोग करके इसे लागू करें

### 8.1.1 विस्तृत कार्यक्रम विशिष्टता और प्रोग्रामिंग भाषा का प्रयोग

कार्यक्रम की विशिष्टता - जब आपको कोई परियोजना दी जाती है, तो कार्यक्रम की विशिष्टता इस बात की परिभाषा होती है कि कार्यक्रम को क्या करना चाहिए। इसमें इसकी डेटा संरचनाओं से लेकर इसके विभिन्न इनपुट और आउटपुट तक सब कुछ शामिल है। उदाहरण के लिए इनपुट लें। सामान्य चश्मा यह होगा कि इनपुट कमांड लाइन या जीयूआई के माध्यम से लिया जाता है या नहीं।

प्रभावों का एक सटीक विवरण जो एक व्यक्तिगत कार्यक्रम को उत्पन्न करना चाहिए। यह स्पष्ट रूप से बताना चाहिए कि यह कैसे करेगा, इस बारे में कोई वादा किए बिना कार्यक्रम क्या करेगा। एक प्रोग्राम के लिए प्रोग्राम विनिर्देश जिसे समाप्त करने का इरादा है, इनपुट-आउटपुट विनिर्देश का रूप ले सकता है जो इनपुट मानों के सेट से आउटपुट मानों के सेट तक वांछित मैपिंग का वर्णन करता है। चक्रीय कार्यक्रमों के लिए एक सरल इनपुट-आउटपुट विनिर्देश प्रदान करना संभव नहीं है जो समाप्त करने का इरादा नहीं है; इसके बजाय, सामान्य अभ्यास कार्यक्रम द्वारा अपने चक्रीय संचालन के दौरान किए गए व्यक्तिगत कार्यों पर ध्यान केंद्रित करना है। टर्मिनेटिंग और चक्रीय दोनों कार्यक्रमों के लिए, प्रोग्राम विनिर्देशों को विभिन्न प्रकार के नोटेशन में लिखा गया है, जिसमें एक प्राकृतिक भाषा से लेकर एम्बेडेड समीकरणों और तालिकाओं के साथ प्रथम-क्रम विधेय कलन के आधार पर औपचारिक संकेतन शामिल हैं।

कई कारणों से, परियोजना में शामिल सभी लोगों के लिए परियोजना का प्रोटोटाइप महत्वपूर्ण है:



चित्र 8.1 प्रोजेक्ट प्रोटोटाइप के कारण

## 8.1.2 बेहतर परियोजना विनिर्देश लिखने के सात तरीके



चित्र 8.2 बेहतर परियोजना विनिर्देश लिखने के सात तरीके:

1. उपयोग के मामले शामिल करें - उपयोग के मामले न केवल एक डेवलपर को यह समझने में मदद करने के लिए महत्वपूर्ण हैं कि कौन सी कार्यक्षमता का निर्माण करना है, बल्कि परियोजना के वांछित परिणाम पर हमें फिर से ध्यान केंद्रित करने के लिए भी महत्वपूर्ण हैं। यह डेवलपर्स के लिए किए जाने वाले कार्यों की सूची नहीं है; बल्कि, यह एक दस्तावेज है जो बताता है कि क्या करने की आवश्यकता है और क्यों। इस पर विचार करें कि उत्पाद कैसे काम करता है लेकिन मानव भाषा में। यह एंड-यूजर से जुड़ने और यह समझने के लिए महत्वपूर्ण है कि वे उत्पाद के साथ कैसे इंटरैक्ट करेंगे।

2. परियोजना विनिर्देशों को साफ-सुथरा व्यवस्थित किया जाना चाहिए - यह सुनिश्चित किया जाना चाहिए कि विनिर्देश पढ़ने में आसान हो। एक परिचय के साथ शुरू करें और सामग्री को तार्किक वर्गों में विभाजित करें। सामग्री की एक तालिका शामिल करें ताकि पाठक भविष्य में अपनी जरूरत के अनुभाग आसानी से ढूंढ सकें।

3. इसे एक जीवित दस्तावेज बनाएं - परियोजना की शुरुआत में आवश्यकताओं को पूरा करने की अपेक्षा करना अवास्तविक है। आवश्यकताओं में परिवर्तन और विकास के रूप में इसे नियमित रूप से अद्यतन करने की आवश्यकता होगी। उपयोगकर्ताओं को इसे हर समय अद्यतित रखने का प्रयास करना चाहिए और यह सुनिश्चित करना चाहिए कि विकास शुरू होने से पहले कुछ भी छूट न जाए। हालाँकि, एक प्रोजेक्ट मैनेजर के रूप में, उपयोगकर्ताओं को यह भी सुनिश्चित करना चाहिए कि क्लाइंट को किसी भी अपडेट के बारे में सूचित किया जाए। यह पाठक को अगले बिंदु पर लाता है।

4. इसे एक औपचारिक दस्तावेज बनाएं - मेरी राय में, क्लाइंट के साथ किसी के प्रोजेक्ट विनिर्देश पर हस्ताक्षर करने की प्रक्रिया औपचारिक रूप से स्थापित की जानी चाहिए। एक परियोजना की योजना के शुरुआती चरणों के दौरान, कई बैठकें, फोन कॉल और दस्तावेजीकरण का आदान-प्रदान होगा। उपयोगकर्ता नहीं चाहते कि उनके प्रोजेक्ट विनिर्देश केवल एक अन्य दस्तावेज बनें जो पढ़ा जाता है लेकिन पूरी तरह से समझ में नहीं आता है।

यह अनुशंसा की जाएगी कि एक बार जब उपयोगकर्ता आवश्यकताओं को लिखना समाप्त कर देता है और एक ग्राहक को समीक्षा करने और प्रतिक्रिया देने का मौका मिलता है, तो उपयोगकर्ता एक पीडीएफ बनाते हैं और इसे क्लाइंट को अनुमोदन के लिए भेजते हैं। आप इसके लिए डॉक्यूमेंटसाइन जैसी सर्विस का भी इस्तेमाल कर सकते हैं। यह क्लाइंट को सचेत करना चाहिए कि परियोजना की आवश्यकता कार्य के सहमत दायरे में हैं, और उन्हें इसका पालन करना चाहिए। यदि इसे किसी भी बिंदु पर बदलने की आवश्यकता है, तो ऐसा किया जा सकता है, लेकिन संशोधित संस्करण पर भी हस्ताक्षर किए जाने चाहिए।

5. अपने औचित्य पर विवरण शामिल करें - परियोजनाएं समय लेने वाली हो सकती हैं - दायरे और परियोजना में देरी के आधार पर, इसे शुरू होने में छह महीने या एक साल भी लग सकता है। न केवल इस बात पर नज़र रखना महत्वपूर्ण है कि क्या निर्णय लिए गए, बल्कि यह भी कि वे क्यों किए गए। इस बारे में विशिष्ट जानकारी शामिल करें कि किसी विशेष पथ को क्यों चुना गया या प्रस्तावित समाधान ही इस परियोजना के लिए काम करने वाला एकमात्र समाधान क्यों हो सकता है। यह न केवल परियोजना के दौरान बल्कि भविष्य में भी बेहद उपयोगी होगा यदि आपको जल्दी से गोता लगाने और यह पता लगाने की आवश्यकता है कि कुछ निर्णय क्यों किए गए थे।

6. जानें कि कब लिखना है - यह कुछ ऐसा है जिसे आप मामला-दर-मामला आधार पर तय करना चाहेंगे। ट्रेलो के निर्माता और एक सॉफ्टवेयर डेवलपर जोएल स्पोल्स्की ने लगभग दो दशक पहले लिखा था कि एक सप्ताह से अधिक कोडिंग कार्य की आवश्यकता वाली किसी भी परियोजना को पूरी तरह से परिभाषित कल्पना के बिना नुकसान होगा। मैं खुद से सहमत होने के इच्छुक हूँ। प्रयोक्ताओं को प्रत्येक प्रोजेक्ट के लिए एक विशिष्ट विवरण की आवश्यकता नहीं होती है, इसलिए उपलब्ध समय और बजट के आधार पर यह निर्णय लें।

अपनी टीम को शामिल करें - प्रोजेक्ट मैनेजर को प्रोजेक्ट स्पेसिफिकेशन बनाने और उसका मालिक बनने का प्रभारी होना चाहिए, लेकिन बहुत सख्त न हों। सुनिश्चित करें कि टीम के डिजाइनर और डेवलपर्स एक साथ काम करते हैं ताकि यह सुनिश्चित हो सके कि विवरण यथासंभव विस्तृत हैं, जब आवश्यक हो तो विस्तृत तकनीकी जानकारी के साथ।

### 8.1.3 प्रोग्रामिंग भाषा

एक प्रोग्रामिंग भाषा एक औपचारिक भाषा है (उल्लेखों का एक सेट जिसे हम एल्गोरिथम नियमों के माध्यम से सटीक रूप से चिह्नित कर सकते हैं) जैसे कि भाषा में एक वाक्य का कम्प्यूटेशनल अर्थ होता है। कंप्यूटर विज्ञान में भाषा या प्रोग्रामिंग भाषा एक कृत्रिम भाषा है जिसका उपयोग निर्देशों के अनुक्रम (एक कंप्यूटर प्रोग्राम) को लिखने के लिए किया जाता है जिसे कंप्यूटर द्वारा चलाया जा सकता है। प्राकृतिक भाषाओं के समान, जैसे अंग्रेजी, प्रोग्रामिंग भाषाओं में एक शब्दावली, व्याकरण और वाक्य रचना होती है। कंप्यूटर प्रोग्राम करने के लिए उपयोग की जाने वाली भाषाओं में सरल तार्किक संरचना होनी चाहिए, और उनके व्याकरण, वर्तनी और विराम चिह्न के नियम सटीक होने चाहिए।

प्रोग्रामिंग भाषा कार्यान्वयन ज्यादातर मामलों में, किसी ऑब्जेक्ट डिजाइन को कोड में परिवर्तित करना एक सरल प्रक्रिया है। C++, Java, Smalltalk, C#, और Python सहित किसी भी ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषा में कक्षाओं का प्रतिनिधित्व किया जा सकता है।

प्रोग्रामिंग लैंग्वेज बनाने के चरण:



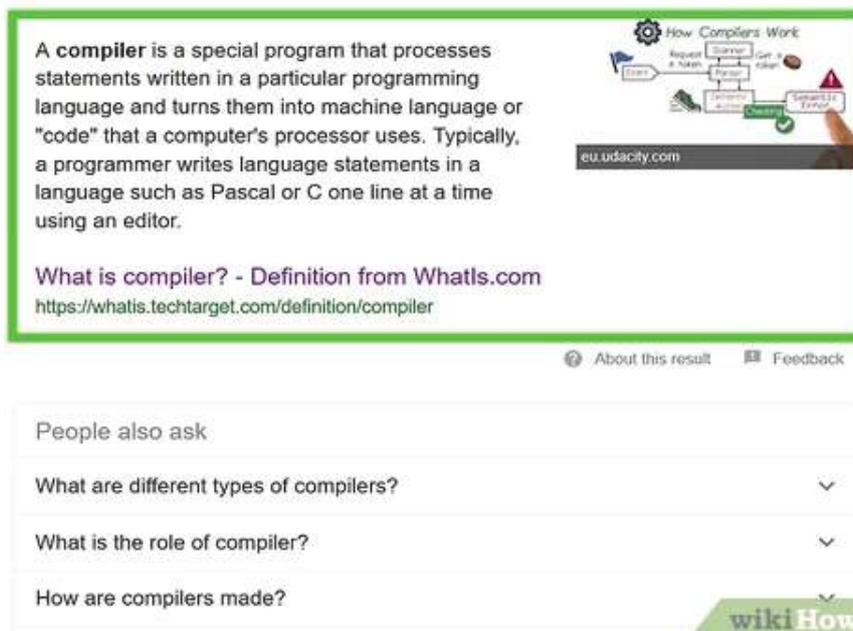
चित्र 8.3 प्रोग्रामिंग भाषा बनाने के चरण

तकनीक से परिचित हों - यदि आप कंप्यूटर का उपयोग करना नहीं जानते हैं तो आप प्रोग्रामिंग भाषा नहीं बना सकते हैं।



चित्र 8.4 तकनीक

शब्दावली से परिचित हों - संकलक लेखक अक्सर अपरिचित शब्दावली का प्रयोग करते हैं। आगे बढ़ने से पहले, कंपाइलर्स के बारे में जानें। सुनिश्चित करें कि आपको जो कुछ भी जानने की जरूरत है उसे सुनिश्चित करें।



चित्र 8.5 शब्दावली

(Image Source: wikihow)

तय करें कि आपकी भाषा किस समस्या का समाधान कर रही है - क्या यह एक डोमेन-विशिष्ट समस्या समाधानकर्ता है या एक सामान्य-उद्देश्य वाली भाषा है?



चित्र 8.6 प्रोग्रामिंग भाषा

(Image Source: wikihow)

4. अपनी भाषा के शब्दार्थ और उसकी अवधारणाओं के बारे में सोचें -

क्या आप डायरेक्ट पॉइंटर एक्सेस की अनुमति देंगे या नहीं?

आपकी भाषा किस प्रकार के डेटा का समर्थन करती है?

क्या यह एक गतिशील या स्थिर भाषा है?

आपके पास कौन सा मेमोरी मॉडल है? क्या आप कचरा संग्रहकर्ता का उपयोग करने जा रहे हैं या अपनी स्मृति को मैनुअल रूप से प्रबंधित करने जा रहे हैं? (यदि आप एक कचरा संग्रहकर्ता का उपयोग करते हैं, तो अपनी भाषा में फिट होने के लिए एक लिखने या मौजूदा को संशोधित करने की योजना बनाएं।)

आप संगामिति को कैसे संभालेंगे? क्या आप लिंडा या अभिनेता मॉडल, या कुछ और जटिल जैसे साधारण थ्रेडिंग/लॉकिंग मॉडल का उपयोग करने जा रहे हैं? (क्योंकि कंप्यूटर में अब कई कोर हैं।)

क्या आदिम कार्यों को भाषा में बनाया गया है, या सब कुछ एक पुस्तकालय से खींचा जाएगा?

आपकी भाषा का प्रतिमान या प्रतिमान क्या है? कार्यात्मक? वस्तु के उन्मुख? प्रोटोटाइप (जावास्क्रिप्ट के समान)? पहलू-उन्मुख? टेम्पलेट-चालित? वैकल्पिक रूप से,

आपकी भाषा मौजूदा पुस्तकालयों और भाषाओं (मुख्य रूप से सी) के साथ कैसे बातचीत करेगी? यदि आप एक डोमेन-विशिष्ट भाषा बना रहे हैं तो यह महत्वपूर्ण है।

अंत में, इन प्रश्नों के कुछ उत्तर दूसरे चरण द्वारा प्रदान किए जाएंगे, जो अगले चरण का उत्तर देने में आपकी सहायता करेंगे।

Contents [hide]	
1	Definitions
2	History
2.1	Early developments
2.2	Refinement
2.3	Consolidation and growth
3	Elements
3.1	Syntax
3.2	Semantics
3.2.1	Static semantics
3.2.2	Dynamic semantics
3.3	Type system
3.3.1	Typed versus untyped languages
3.3.2	Static versus dynamic typing
3.3.3	Weak and strong typing
3.4	Standard library and run-time system
4	Design and implementation
4.1	Specification
4.2	Implementation
5	Proprietary languages
6	Use
6.1	Measuring language usage
7	Dialects, flavors and implementations
8.	Taxonomies

wikiHow

चित्र 8.7 भाषा और अवधारणाओं के शब्दार्थ

(Image Source: wikiHow)

कुछ विशिष्ट कार्यों के बारे में सोचें जो कोई आपकी भाषा के साथ करने में सक्षम होना चाहेगा - उदाहरण के लिए, 'वे रोबोट को एक पंक्ति का अनुसरण करने के लिए निर्देशित करना चाहते हैं,' या 'वे इसमें अपेक्षाकृत पोर्टेबल डेस्कटॉप प्रोग्राम बनाना चाहते हैं,' या 'हो सकता है कि वे इसके साथ वेब अनुप्रयोग बनाना चाहें।'

## Standardization [\[ edit \]](#)

Not all major programming languages have specifications, and languages can exist and be popular for *facto* standard, without this behavior being documented in a specification. Perl (through Perl 5) is a notable example.<sup>[1]</sup> A language may be implemented and then specified, or specified and then implemented, or they may check on each other: writing a specification requires precisely stating the behavior of an implementation, and an implementation has largely been avoided since ALGOL 68 (1968), due to unexpected difficulties in its popularity without a formal specification: an implementation is essential for use, while a specification is

“ ALGOL 68 was the first (and possibly one of the last) major lan

## Forms [\[ edit \]](#)

A programming language specification can take several forms, including the following:

- An explicit definition of the **syntax** and **semantics** of the language. While syntax is commonly specified (e.g., the C language), or a formal semantics (e.g., the Standard ML<sup>[3]</sup> and Scheme<sup>[4]</sup> specifications). A notable example is the book, *The C Programming Language* (1978), and only much later being formally standardized.
- A description of the behavior of a compiler (sometimes called "translator") for the language (e.g., the C compiler, which may be written in natural or a formal language.
- A *model* implementation, sometimes written in the language being specified (e.g., the C compiler, which may be written in natural or a formal language.

चित्र 8.8 भाषा के साथ प्रदर्शन करने के लिए कार्य

(Image Source: wikihow)

वाक्य रचना विचारों के साथ प्रयोग (भाषा का पाठ) - अपनी सामग्री को संदर्भ-मुक्त भाषा श्रेणी या उसके भीतर कुछ रखें। आप बाद में अपने पार्सर जनरेटर की सराहना करेंगे।

## Syntax [\[ edit \]](#)

This section **needs expansion**. You can help by adding to it. (February 2018)

The **syntax** of a programming language is usually described using a combination of the following two components:

- a regular expression describing its lexemes, and
- a context-free grammar which describes how lexemes may be combined to form a syntactically correct program.

## Semantics [\[ edit \]](#)

Formulating a rigorous semantics of a large, complex, practical programming language is a daunting task even for experienced programmers. The following are some of the ways in which programming language semantics can be described, all in natural language:

- **Natural language:** Description by human natural language.
- **Formal semantics:** Description by mathematics.
- **Reference implementations:** Description by computer program.
- **Test suites:** Description by examples of programs and their expected behaviors. While few language specifications include test suites, the semantics of a test suite (e.g. in the past the specification of Ada has been modified to match the behavior of the reference implementation).

## Natural language [\[ edit \]](#)

Most widely used languages are specified using natural language descriptions of their semantics. This description uses natural language, e.g., the print version of *The Java Language Specification, 3rd Ed.* is 596 pages long.

The imprecision of natural language as a vehicle for describing programming language semantics can lead to problems. For example, the specification of the C language in English, and it was later discovered that the specification did not provide adequate guidance for implementations.<sup>[6]</sup>

चित्र 8.9 सिंटैक्स विचार

(Image Source: wikihow)

वाक्य रचना के लिए एक औपचारिक व्याकरण लिखें - वाक्य रचना के लिए एक औपचारिक व्याकरण लिखें

Metacharacter	
<code>^</code>	Matches the starting position within the string. In line-based tools, it matches the starting position of the line.
<code>.</code>	Matches any single character (many applications exclude newlines, and exactly which characters that the line feed character is included). Within POSIX bracket expressions, the dot character is treated as a literal character.
<code>[ ]</code>	A bracket expression. Matches a single character that is contained within the brackets. For example, <code>[abc]</code> matches "a", "b", or "c". These forms can be mixed: <code>[abcx-z]</code> matches "a", "b", "c", "x", "y", or "z". The <code>-</code> character is treated as a literal character if it is the last or the first (after the <code>^</code> , if present) character in a bracket expression. If it is the first (after the <code>^</code> ) character, it can be included in a bracket expression if it is the first (after the <code>^</code> ) character.
<code>[^ ]</code>	Matches a single character that is not contained within the brackets. For example, <code>[^abc]</code> matches any lowercase letter from "a" to "z". Likewise, literal characters and ranges can be mixed.
<code>\$</code>	Matches the ending position of the string or the position just before a string-ending newline.
<code>( )</code>	Defines a marked subexpression. The string matched within the parentheses can be recalled with <code>\1</code> . <b>mode requires <code>\( \)</code>.</b>
<code>\n</code>	Matches what the <i>n</i> th marked subexpression matched, where <i>n</i> is a digit from 1 to 9. This can be used to match groups.
<code>*</code>	Matches the preceding element zero or more times. For example, <code>ab*c</code> matches "ac", "abc", "abab", "ababab", and so on.
<code>{m,n}</code>	Matches the preceding element at least <i>m</i> and not more than <i>n</i> times. For example, <code>a{3,5}</code> matches "aaa", "aaaa", and "aaaaa". <b>mode requires <code>\{m,n\}</code>.</b>

चित्र 8.10 औपचारिक व्याकरण

(Image Source: wikiHow)

7. तय करें कि भाषा की व्याख्या या संकलन किया जाएगा या नहीं -

**Examples:**

- `.at` matches any three-character string ending with "at", including "hat", "cat", and "bat".
- `[hc]at` matches "hat" and "cat".
- `[^b]at` matches all strings matched by `.at` except "bat".
- `[^hc]at` matches all strings matched by `.at` other than "hat" and "cat".
- `^[hc]at` matches "hat" and "cat", but only at the beginning of the string or line.
- `[hc]at$` matches "hat" and "cat", but only at the end of the string or line.
- `\[ \]` matches any single character surrounded by "[" and "]" since the brackets are escaped, for example, `\[a\]` matches "[a]".
- `s.*` matches *s* followed by zero or more characters, for example: "s" and "saw" and "seed".

**POSIX extended** [\[ edit \]](#)

The meaning of metacharacters escaped with a backslash is reversed for some characters in the POSIX standard. So, for example, `\( \)` is now `( )` and `\{ \}` is now `{ }`. Additionally, `\?` is now `?`.

Metacharacter	
<code>?</code>	Matches the preceding element zero or one time. For example, <code>ab?c</code> matches only "abc".
<code>+</code>	Matches the preceding element one or more times. For example, <code>ab+c</code> matches "abc", "abbc", "abbbc", and so on.
<code> </code>	The choice (also known as alternation or set union) operator matches either the expression to its left or the expression to its right. For example, <code>a b</code> matches "a" or "b".

**Examples:** [wikiHow to Create a Programming Language](#)

चित्र 8.11 व्याख्या या संकलित

(Image Source: wikiHow)

फ्रंटएंड स्कैनर और पार्सर लिखें या ऐसा टूल ढूँढ़ें जो इसमें आपकी मदद करे - विचार करें कि आपका कंपाइलर/दुभाषिया आपके उपयोगकर्ता को गलत प्रोग्राम और सिंटैक्स त्रुटियों के बारे में कैसे सूचित करेगा।

POSIX	Non-standard	Perl/Tcl	Vim	Java	ASCII
	<code>[::ascii:]</code> [29]			<code>\p{ASCII}</code>	<code>[\x00-\x7F]</code>
<code>[::alnum:]</code>				<code>\p{Alnum}</code>	<code>[A-Za-z0-9]</code>
	<code>[::word:]</code> [29]	<code>\w</code>	<code>\w</code>	<code>\w</code>	<code>[A-Za-z0-9_]</code>
		<code>\W</code>	<code>\W</code>	<code>\W</code>	<code>[^A-Za-z0-9_]</code>
<code>[::alpha:]</code>			<code>\a</code>	<code>\p{Alpha}</code>	<code>[A-Za-z]</code>
<code>[::blank:]</code>			<code>\s</code>	<code>\p{Blank}</code>	<code>[ \t]</code>
		<code>\b</code>	<code>\&lt; \&gt;</code>	<code>\b</code>	<code>(?c=\W) (?=\w)   (?c=\w) (?=\W)</code>
				<code>\B</code>	<code>(?c=\W) (?=\W)   (?c=\w) (?=\w)</code>
<code>[::cntrl:]</code>				<code>\p{Cntrl}</code>	<code>[\x00-\x1F\x7F]</code>
<code>[::digit:]</code>		<code>\d</code>	<code>\d</code>	<code>\p{Digit}</code> or <code>\d</code>	<code>[0-9]</code>
		<code>\D</code>	<code>\D</code>	<code>\D</code>	<code>[^0-9]</code>
<code>[::graph:]</code>				<code>\p{Graph}</code>	<code>[\x21-\x7E]</code>
<code>[::lower:]</code>			<code>\l</code>	<code>\p{Lower}</code>	<code>[a-z]</code>
<code>[::print:]</code>			<code>\p</code>	<code>\p{Print}</code>	<code>[\x20-\x7E]</code>
<code>[::punct:]</code>				<code>\p{Punct}</code>	<code>[!\"#\$%&amp;'()*+,-./:;&lt;=&gt;?@[\^_`{ }~--</code>
<code>[::space:]</code>		<code>\s</code>	<code>\_s</code>	<code>\p{Space}</code> or <code>\s</code>	<code>[ \t\r\n\v\f]</code>
		<code>\S</code>	<code>\_S</code>	<code>\S</code>	<code>[^\t\r\n\v\f]</code>
<code>[::upper:]</code>			<code>\u</code>	<code>\p{Upper}</code>	<code>[A-Z]</code>
<code>[::xdigit:]</code>			<code>\x</code>	<code>\p{XDigit}</code>	<code>[A-Fa-f0-9]</code>

चित्र 8.12 फ्रंटेंड स्कैनर और पार्सर

(Image Source: wikihow)

8. ऑब्जेक्ट कोड या मध्यवर्ती प्रतिनिधित्व लिखने के लिए पार्सर जानकारी का उपयोग करें - एएसटी तीन पता कोड या उसके बड़े भाई एएसएसए के साथ, फिर अपने कार्यों, वैश्विक चर, आदि को परिभाषित करने के लिए एक प्रतीक तालिका बनाएं।

अपनी भाषा के आधार पर, आप अपनी कक्षाओं के लिए (प्रतिबिंब या RATTI का समर्थन करने के लिए) वर्चुअल पॉइंटर टेबल या सूचना तालिकाएँ बनाना चाह सकते हैं।

```

Sstring1 = "Hello World\n";
if ($string1 =~ m/...../) {
    print "Sstring1 has length >= 5.\n";
}

Output:

Hello World
has length >= 5.

Sstring1 = "Hello World\n";
if ($string1 =~ m/(H..)(o..)/) {
    print "We matched '$1' and '$2'.\n";
}

Output:

We matched 'Hel' and 'o W'.
    
```

चित्र 8.13 ऑब्जेक्ट कोड लिखने के लिए पार्सर जानकारी (Image Source: wikihow)

8. निष्पादक या कोड जनरेटर लिखें जो सब कुछ एक साथ बांध देगा -

```

$string1 = "Hello World\n";
if ($string1 =~ m/1{1,2}/) {
    print "There exists a substring with at least 1 ";
    print "and at most 2 1's in $string1\n";
}

Output:

There exists a substring with at least 1 and at most 2 1's in
Hello World

$string1 = "Hello World\n";
if ($string1 =~ m/[aeiou]+/) {
    print "$string1 contains one or more vowels.\n";
}

Output:
    
```

wikiHow

चित्र 8.14 निष्पादक या कोड जनरेटर

(Image Source: wikiHow)

8. भाषा का परीक्षण करने के लिए कई परीक्षण कार्यक्रम लिखें - आप ऐसे प्रोग्राम लिखना चाहते हैं जो आपके औपचारिक व्याकरण की बाधाओं पर जोर दें ताकि यह सुनिश्चित हो सके कि आपका कंपाइलर आपकी परिभाषा के भीतर सब कुछ स्वीकार करता है और इसके बाहर की हर चीज को अस्वीकार करता है।

```

Matches a zero-width boundary between a word-class character (see next) and either a non-word class
character or an edge, same as.
(?\w\W|\W\w|\b|B)

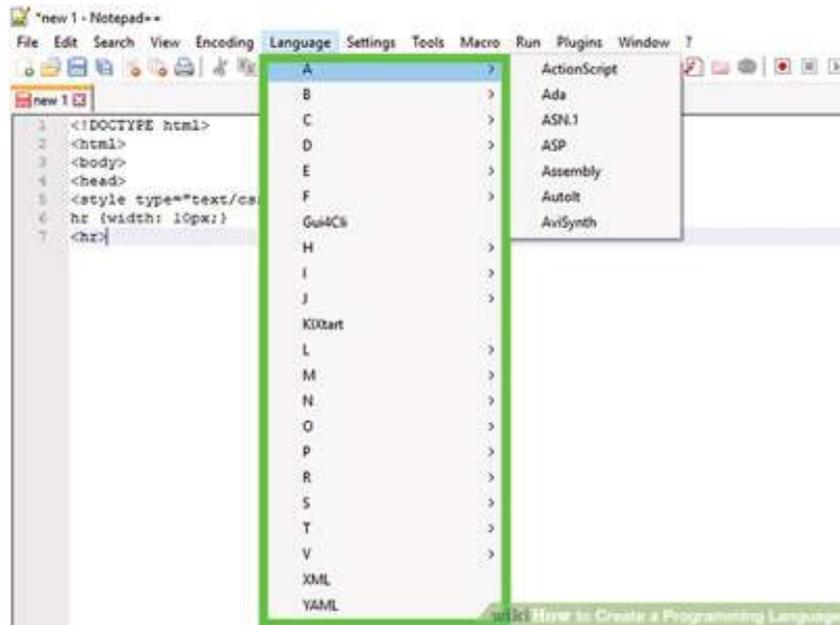
Matches an alphanumeric character, including "_":
same as [A-Za-z0-9_] in ASCII, and
[\p{Alphabetic}\p{GC=Mark}\p{GC=Decimal_Number}\p{GC=Connector_Function}]
in Unicode[7] where the Alphabetic property contains more than Latin letters, and the
Decimal_Number property contains more than Arab digits.
    
```

wikiHow

चित्र 8.15 भाषा का परीक्षण करने के लिए परीक्षण कार्यक्रम

(Image Source: wikiHow)

विचार करें कि उपयोगकर्ता अपने कार्यक्रमों को कैसे डीबग करेगा -



चित्र 8.16 प्रोग्रामों को डीबग करें

(Image Source: wikihow)

10. यदि आपकी भाषा एक मानक पुस्तकालय का उपयोग करती है, तो आप इसे लिखना चाहेंगे - यदि आवश्यक हो तो कचरा संग्रहकर्ता और अन्य रनटाइम सुविधाओं के साथ। विशेष रूप से, यदि आप एक कंपाइलर लिखते हैं, तो आपको उस कोड की आवश्यकता होगी जो ऑपरेटिंग सिस्टम उपयोगकर्ता कोड को चलाने के लिए निष्पादित करेगा (उदाहरण के लिए, सभी वैश्विक चर आवंटित करना)।

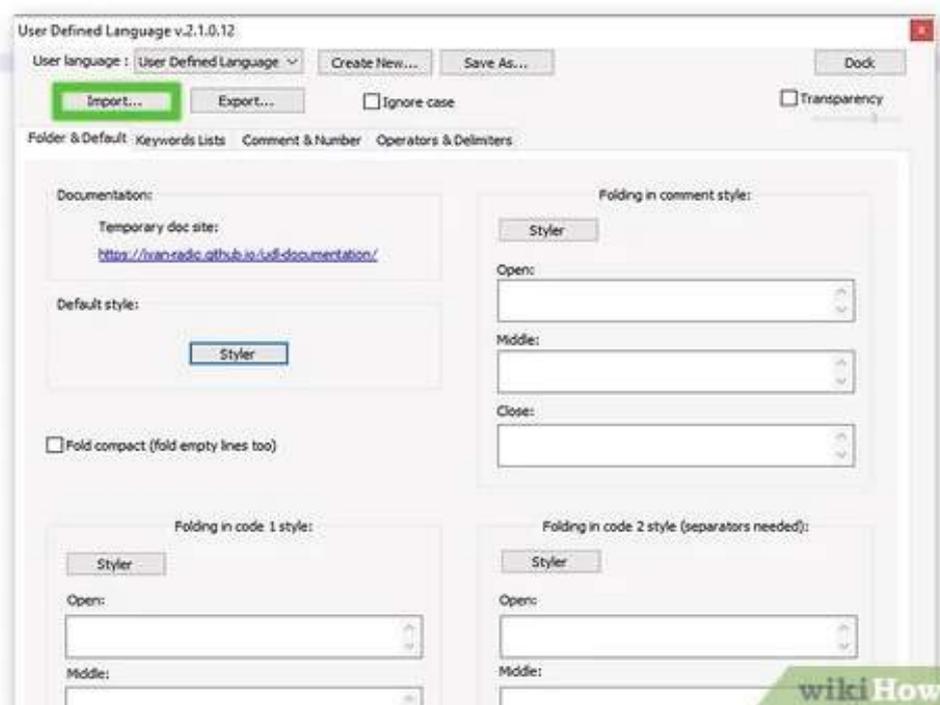
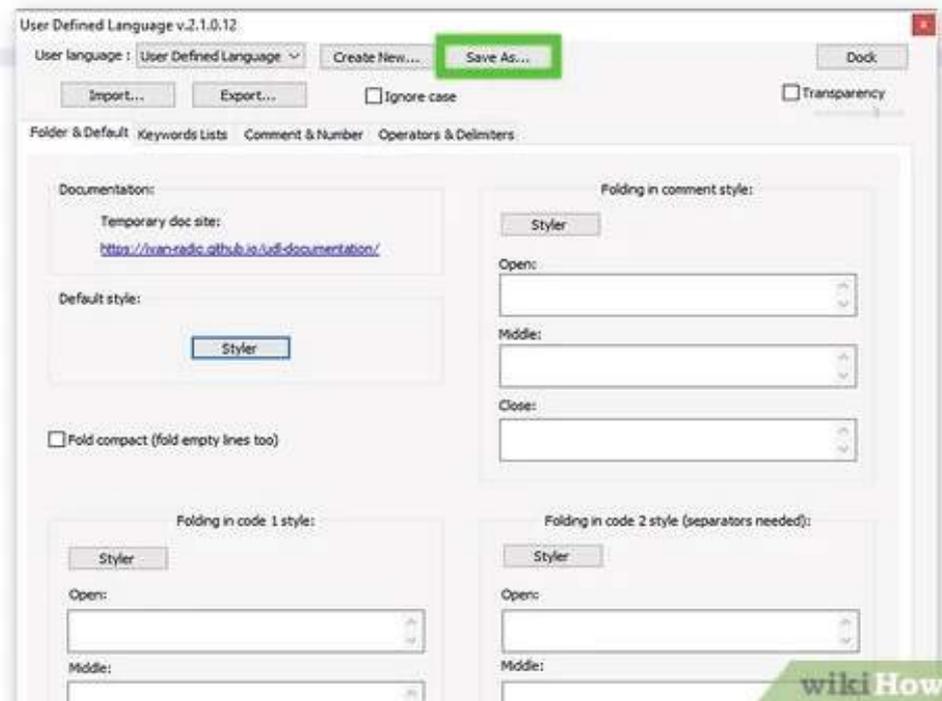


Fig 8.17 Standard Library

अपनी भाषा को इसके विनिर्देशों के साथ प्रकाशित करें और इसमें आप क्या कर सकते हैं इसके कुछ उदाहरण - मौजूदा पुस्तकालयों और भाषाओं के साथ एकीकृत करने के तरीके के साथ-साथ रनटाइम सुविधाओं और/या मानक का उपयोग करने के तरीके को दस्तावेज़ करना न भूलें पुस्तकालय।



चित्र 8.18 भाषा प्रकाशित करें  
(Image Source: wikihow)

## सारांश

- कार्यक्रम की विशिष्टता - जब आपको कोई परियोजना दी जाती है, तो कार्यक्रम की विशिष्टता इस बात की परिभाषा होती है कि कार्यक्रम को क्या करना चाहिए। इसमें इसकी डेटा संरचनाओं से लेकर इसके विभिन्न इनपुट और आउटपुट तक सब कुछ शामिल है।
- एक प्रोग्रामिंग भाषा एक औपचारिक भाषा है (उल्लेखों का एक सेट जिसे हम एल्गोरिथम नियमों के माध्यम से सटीक रूप से चिह्नित कर सकते हैं) जैसे कि भाषा में एक वाक्य का कम्प्यूटेशनल अर्थ होता है।
- कंप्यूटर विज्ञान में भाषा या प्रोग्रामिंग भाषा एक कृत्रिम भाषा है जिसका उपयोग निर्देशों के अनुक्रम (एक कंप्यूटर प्रोग्राम) को लिखने के लिए किया जाता है जिसे कंप्यूटर द्वारा चलाया जा सकता है। प्राकृतिक भाषाओं के समान, जैसे अंग्रेजी, प्रोग्रामिंग भाषाओं में एक शब्दावली, व्याकरण और वाक्य रचना होती है।
- प्रोग्रामिंग भाषा कार्यान्वयन ज्यादातर मामलों में, किसी ऑब्जेक्ट डिज़ाइन को कोड में परिवर्तित करना एक सरल प्रक्रिया है। C++, Java, Smalltalk, C#, और Python सहित किसी भी ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषा में कक्षाओं का प्रतिनिधित्व किया जा सकता है।
- शब्दावली से परिचित हों - संकलक लेखक अक्सर अपरिचित शब्दावली का प्रयोग करते हैं। आगे बढ़ने से पहले, कंपाइलर्स के बारे में जानें। सुनिश्चित करें कि आपको जो कुछ भी जानने की जरूरत है उसे सुनिश्चित करें।
- अपनी भाषा को इसके विनिर्देशों के साथ प्रकाशित करें और इसमें आप क्या कर सकते हैं इसके कुछ उदाहरण - मौजूदा पुस्तकालयों और भाषाओं के साथ एकीकृत करने के तरीके के साथ-साथ रनटाइम सुविधाओं और/या मानक पुस्तकालयों का उपयोग करने के तरीके को दस्तावेज करना न भूलें।
- ऑब्जेक्ट कोड या मध्यवर्ती प्रतिनिधित्व लिखने के लिए पार्सर जानकारी का उपयोग करें - एएसटी तीन पता कोड या उसके बड़े भाई एसएसए के साथ, फिर अपने कार्यों, वैश्विक चर, आदि को परिभाषित करने के लिए एक प्रतीक तालिका बनाएं।
- अपनी भाषा के आधार पर, आप अपनी कक्षाओं के लिए वर्चुअल पॉइंटर टेबल या सूचना तालिका भी बनाना चाह सकते हैं (प्रतिबिंब या आरटीटीआई का समर्थन करने के लिए)।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

प्रोग्रामिंग लैंग्वेज क्या है?

प्रोग्रामिंग भाषा बनाने के चरण क्या हैं?

कार्यक्रम विनिर्देशों को परिभाषित करें।

रिक्त स्थानों की पूर्ति कीजिए :

एक प्रोग्रामिंग भाषा एक \_\_\_\_\_ है (उल्लेखों का एक सेट जिसे हम एल्गोरिथम नियमों के माध्यम से सटीक रूप से चिह्नित कर सकते हैं) जैसे कि भाषा में एक वाक्य का कम्प्यूटेशनल अर्थ होता है।

\_\_\_\_\_ इस बात की परिभाषा है कि प्रोग्राम को क्या करना चाहिए। इसमें इसकी डेटा संरचनाओं से लेकर इसके विभिन्न इनपुट और आउटपुट तक सब कुछ शामिल है।

बेहतर परियोजना विनिर्देश लिखने के सात तरीके क्या हैं?

## यूनिट 8.2: अनुप्रयोगों और प्रमुख प्रक्रियाओं को विकसित करने के लिए दृष्टिकोण

### यूनिट के उद्देश्य

इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. अनुप्रयोगों और प्रमुख प्रक्रियाओं को विकसित करने के लिए विभिन्न सॉफ्टवेयर इंजीनियरिंग दृष्टिकोणों की पहचान करें

### 8.2.1 सॉफ्टवेयर इंजीनियरिंग के लिए विभिन्न दृष्टिकोण

सॉफ्टवेयर विश्लेषण और डिजाइन के लिए दो मुख्य तकनीकें हैं:



चित्र 8.19 सॉफ्टवेयर इंजीनियरिंग के लिए दृष्टिकोण

1. फंक्शन ओरिएंटेड डिजाइन: प्रोग्राम के फंक्शन पर फोकस करने का नतीजा फंक्शन ओरिएंटेड डिजाइन है। यह चरणबद्ध शोधन की शोधन पद्धति पर आधारित है। चरणबद्ध शोधन करने के लिए पुनरावृत्त प्रक्रियात्मक अपघटन का उपयोग किया जाता है। स्टेपवाइज रिफाइनमेंट प्रोग्राम रिफाइनमेंट के लिए एक टॉप-डाउन दृष्टिकोण है जिसमें एक प्रोग्राम को विस्तार के बढ़ते स्तरों के पदानुक्रम में परिष्कृत किया जाता है। हम कार्यक्रम के कार्यों के उच्च-स्तरीय विवरण के साथ शुरू करते हैं। फिर, प्रत्येक चरण में, हम अपने उच्च-स्तरीय विवरण के एक हिस्से को परिष्कृत करते हैं। विस्तार शोधन की प्रक्रिया है। दृष्टिकोण एक उच्च-स्तरीय वैचारिक मॉडल के साथ शुरू होना चाहिए और निचले स्तर की बारीकियों तक अपना काम करना चाहिए। प्रत्येक मॉड्यूल को तब तक परिष्कृत किया जाता है जब तक कि हमारी प्रोग्रामिंग भाषा का कथन स्तर नहीं पहुंच जाता।

2. ऑब्जेक्ट-ओरिएंटेड डिजाइन: ऑब्जेक्ट-ओरिएंटेड डिजाइन प्रोग्राम द्वारा किए गए फंक्शन के बजाय प्रोग्राम द्वारा संशोधित किए जाने वाले डेटा पर ध्यान केंद्रित करने का परिणाम है। नतीजतन, यह फंक्शन-ओरिएंटेड डिजाइन का विरोध करता है। ऑब्जेक्ट-ओरिएंटेड डिजाइन वास्तविक दुनिया में "चीजों" को देखने से शुरू होता है। व्यक्तिगत रूप से, इन संस्थाओं की विशेषताओं और व्यवहार के संदर्भ में विशेषताएं हैं। ऑब्जेक्ट स्व-निहित इकाइयाँ हैं जिन्हें आसानी से बदला जा सकता है क्योंकि सभी राज्य और प्रतिनिधित्व की जानकारी वस्तु के भीतर संग्रहीत होती है। वस्तुओं को क्रमिक या समवर्ती रूप से वितरित और निष्पादित किया जा सकता है। ऑब्जेक्ट-ओरिएंटेड टेक्नोलॉजी में तीन कीवर्ड शामिल हैं -



चित्र 8.20 वस्तु-उन्मुख प्रौद्योगिकी कीवर्ड

### 8.2.2 विकास के दृष्टिकोण

परियोजना जीवन चक्र के दौरान उत्पाद, सेवा या परिणाम बनाने और विकसित करने के लिए उपयोग की जाने वाली विधि, जैसे:



चित्र 8.21 डेवलपमेंट के दृष्टिकोण

**प्रेडिक्टिव** - प्रेडिक्टिव डेवलपमेंट एक सॉफ्टवेयर डेवलपमेंट मेटड है जिसमें मॉडल को डिज़ाइन, कार्यान्वित और परीक्षण किया जाता है, जब तक कि उत्पाद पूरा नहीं हो जाता है - या सभी आवश्यकताओं को पूरा नहीं किया जाता है।

**पुनरावृत्ति** - पुनरावृत्ति एक सॉफ्टवेयर डेवलपमेंट विधि है जो एक बड़े एप्लिकेशन को छोटे चरणों में बनाने की प्रक्रिया को विभाजित करती है। प्रत्येक "पुनरावृत्ति" संपूर्ण विकास प्रक्रिया का प्रतिनिधित्व करता है और इसमें योजना, डिजाइन, विकास और परीक्षण के चरण शामिल हैं।

फुर्तीली - फुर्तीली सॉफ्टवेयर विकास पुनरावृत्ति सॉफ्टवेयर विकास पद्धतियों के एक सेट को संदर्भित करता है जिसमें स्वयं-संगठित क्रॉस-फ़ंक्शनल टीमों के बीच सहयोग के माध्यम से आवश्यकताएं और समाधान विकसित होते हैं।

वृद्धिशील - सॉफ्टवेयर इंजीनियरिंग में, वृद्धिशील विकास एक प्रक्रिया पद्धति है जो लक्ष्य की ओर छोटे कदम उठाने के महत्व पर जोर देती है।

हाइब्रिड - एक हाइब्रिड पद्धति का लक्ष्य जलप्रपात दृष्टिकोण के साथ आवश्यकताओं को अग्रिम (योजना, बजट, आदि) को परिभाषित करने में सक्षम होना है, और फिर डिजाइन, विकास और परीक्षण के लिए एक चुस्त दृष्टिकोण के लिए संक्रमण करना है। बेहतरीन फुर्तीले और जलप्रपात को मिला दिया जाता है और एक संकर में एक साथ काम करने के लिए बनाया जाता है।

### 8.2.3 सॉफ्टवेयर इंजीनियरिंग के विभिन्न अनुप्रयोग

निम्नलिखित चित्र सॉफ्टवेयर के विभिन्न अनुप्रयोगों को सूचीबद्ध करता है:



चित्र 8.22 सॉफ्टवेयर के विभिन्न अनुप्रयोग

1. सिस्टम सॉफ्टवेयर: कंप्यूटर संसाधनों के प्रबंधन और एप्लिकेशन प्रोग्राम निष्पादन का समर्थन करने के लिए सिस्टम सॉफ्टवेयर की आवश्यकता होती है। इस श्रेणी में ऑपरेटिंग सिस्टम, कंपाइलर, संपादक और ड्राइवर जैसे सॉफ्टवेयर शामिल हैं। ये कंप्यूटर के कार्य करने के लिए आवश्यक हैं। किसी प्रोग्राम की मशीन पर निर्भर जरूरतों को उस मशीन की क्षमताओं से जोड़ने के लिए ऑपरेटिंग सिस्टम की आवश्यकता होती है, जिस पर वह चलता है। कंपाइलर प्रोग्राम को उच्च-स्तरीय से मशीनी भाषा में परिवर्तित करते हैं।
2. नेटवर्किंग और वेब एप्लिकेशन सॉफ्टवेयर: नेटवर्किंग सॉफ्टवेयर कंप्यूटर को एक दूसरे के साथ संचार करने और डेटा स्टोरेज सुविधाओं के साथ आवश्यक समर्थन प्रदान करता है। जब सॉफ्टवेयर कंप्यूटर के नेटवर्क पर चल रहा होता है, तो नेटवर्किंग सॉफ्टवेयर का भी उपयोग किया जाता है (जैसे वर्ल्ड वाइड वेब)। इसमें सभी नेटवर्क प्रबंधन सॉफ्टवेयर, सर्वर सॉफ्टवेयर, सुरक्षा और एन्क्रिप्शन सॉफ्टवेयर, और वेब-विकास सॉफ्टवेयर जैसे HTML, PHP, XML, आदि शामिल हैं।
3. एंबेडेड सॉफ्टवेयर: इस प्रकार का सॉफ्टवेयर हार्डवेयर में एम्बेडेड होता है, आमतौर पर रीड-ओनली मेमोरी (ROM) में, एक बड़े सिस्टम के हिस्से के रूप में, और नियंत्रण स्थितियों के तहत विशिष्ट कार्यक्षमता का समर्थन करने के लिए उपयोग किया जाता है। वॉशिंग मशीन, सैटेलाइट, माइक्रोवेव आदि जैसे इंस्ट्रूमेंटेशन और कंट्रोल एप्लिकेशन में इस्तेमाल किया जाने वाला सॉफ्टवेयर एक उदाहरण है।

4. आरक्षण सॉफ्टवेयर: एक आरक्षण प्रणाली का उपयोग प्राथमिक रूप से सूचनाओं को संग्रहीत करने और पुनः प्राप्त करने के साथ-साथ हवाई यात्रा, कार किराए पर लेने, होटल और अन्य गतिविधियों से संबंधित लेनदेन करने के लिए किया जाता है। वे बस और ट्रेन आरक्षण भी प्रदान करते हैं, हालांकि ये हमेशा मुख्य प्रणाली के साथ एकीकृत नहीं होते हैं। इनका उपयोग होटल उद्योग में उपयोगकर्ताओं को कम्प्यूटरीकृत जानकारी देने के लिए भी किया जाता है, जैसे आरक्षण करना और यह सुनिश्चित करना कि होटल ओवरबुक नहीं है।
5. बिजनेस सॉफ्टवेयर: यह सॉफ्टवेयर की सबसे अधिक इस्तेमाल की जाने वाली श्रेणी है और इसका उपयोग व्यावसायिक अनुप्रयोगों का समर्थन करने के लिए किया जाता है। उदाहरणों में इन्वेंट्री मैनेजमेंट सॉफ्टवेयर, अकाउंटिंग सॉफ्टवेयर, बैंकिंग सॉफ्टवेयर, अस्पताल, स्कूल, स्टॉक एक्सचेंज आदि शामिल हैं।
6. मनोरंजन सॉफ्टवेयर: शिक्षा और मनोरंजन सॉफ्टवेयर शैक्षिक संगठनों के लिए एक शक्तिशाली उपकरण है, विशेष रूप से वे जो छोटे बच्चों के साथ काम करते हैं। कम्प्यूटर गेम, शैक्षिक खेल, अनुवाद सॉफ्टवेयर, मैपिंग सॉफ्टवेयर और मनोरंजन सॉफ्टवेयर के अन्य रूप उपलब्ध हैं।
7. आर्टिफिशियल इंटेलिजेंस सॉफ्टवेयर: इस श्रेणी में सॉफ्टवेयर जैसे विशेषज्ञ सिस्टम, डिजीजन सपोर्ट सिस्टम, पैटर्न रिकग्निशन सॉफ्टवेयर, आर्टिफिशियल न्यूरल नेटवर्क आदि शामिल हैं। उनमें जटिल समस्याएं शामिल हैं जो गैर-संख्यात्मक एल्गोरिदम के साथ की गई जटिल गणनाओं से अप्रभावित हैं।
8. वैज्ञानिक सॉफ्टवेयर: वैज्ञानिक और इंजीनियरिंग सॉफ्टवेयर उद्यम-विशिष्ट कार्यों को करने के लिए वैज्ञानिक या इंजीनियरिंग उपयोगकर्ता की आवश्यकता को पूरा करते हैं। ऐसा सॉफ्टवेयर क्षेत्र-विशिष्ट सिद्धांतों, तकनीकों और सूत्रों का उपयोग करके विशिष्ट अनुप्रयोगों के लिए लिखा जाता है। MATLAB, AUTOCAD, PSPICE, ORCAD, और अन्य जैसे सॉफ्टवेयर इसके उदाहरण हैं।

### 8.2.4 डीबीएमएस (DBMS)

डेटाबेस प्रबंधन प्रणाली (डीबीएमएस के रूप में भी जाना जाता है) उपयुक्त सुरक्षा उपायों पर विचार करके उपयोगकर्ताओं के डेटा को संग्रहीत करने और पुनर्प्राप्त करने के लिए सॉफ्टवेयर है। यह उपयोगकर्ताओं को उनकी आवश्यकताओं के अनुसार अपने डेटाबेस बनाने की अनुमति देता है।

इसमें प्रोग्रामों का एक समूह होता है जो डेटाबेस में हेरफेर करता है और डेटाबेस के बीच एक इंटरफ़ेस प्रदान करता है। इसमें डेटाबेस और अन्य एप्लिकेशन प्रोग्राम का उपयोग शामिल है।

DBMS किसी एप्लिकेशन से डेटा के लिए अनुरोध स्वीकार करता है और ऑपरेटिंग सिस्टम को विशिष्ट डेटा प्रदान करने का निर्देश देता है। बड़ी प्रणालियों में, एक DBMS उपयोगकर्ताओं और अन्य तृतीय-पक्ष सॉफ्टवेयर को डेटा संग्रहीत करने और पुनर्प्राप्त करने में मदद करता है।

डेटाबेस प्रबंधन प्रणाली की विशेषताएं:



चित्र 8.23 डीबीएमएस के लक्षण

## 8.2.5 डीबीएमएस के अनुप्रयोग और उपयोग

DBMS के अनुप्रयोग और उपयोग निम्नलिखित हैं:

### 1. रेलवे आरक्षण प्रणाली

टिकट बुकिंग, ट्रेन प्रस्थान और आगमन की स्थिति का रिकॉर्ड रखने के लिए डेटाबेस की आवश्यकता होती है। इसके अलावा अगर ट्रेनें लेट हो जाती हैं तो लोगों को डेटाबेस अपडेट के जरिए इसका पता चल जाता है।



चित्र 8.24 रेलवे आरक्षण प्रणाली

### 2. पुस्तकालय प्रबंधन प्रणाली

पुस्तकालय में हजारों पुस्तकें हैं इसलिए सभी पुस्तकों का रिकॉर्ड एक कॉपी या रजिस्टर में रखना बहुत मुश्किल है। इसलिए डीबीएमएस का उपयोग बुकिंग जारी करने की तारीखों, पुस्तक का नाम, लेखक और पुस्तक की उपलब्धता से संबंधित सभी सूचनाओं को बनाए रखने के लिए किया जाता है।



चित्र 8.25 पुस्तकालय प्रबंधन प्रणाली

### 3. बैंकिंग

हम रोजाना बैंकों के माध्यम से हजारों लेन-देन करते हैं और बिना बैंक जाए हम ऐसा कर सकते हैं। तो कैसे बैंकिंग इतनी आसान हो गई है कि घर बैठे हम बैंकों के माध्यम से पैसा भेज या प्राप्त कर सकते हैं। यह सब सिर्फ डीबीएमएस की वजह से संभव है जो सभी बैंक लेनदेन का प्रबंधन करता है।



चित्र 8.26 बैंकिंग

#### 4. विश्वविद्यालय और कॉलेज

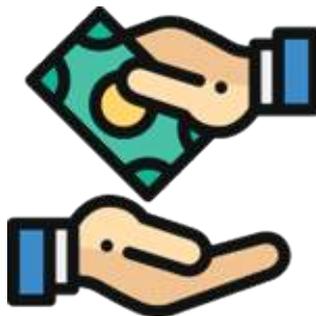
आज परीक्षाएं ऑनलाइन होती हैं और विश्वविद्यालय और कॉलेज डीबीएमएस के माध्यम से इन सभी रिकॉर्ड को बनाए रखते हैं। छात्र पंजीकरण विवरण, परिणाम, पाठ्यक्रम और ग्रेड सभी जानकारी एक डेटाबेस में संग्रहीत की जाती है।



चित्र 8.27 विश्वविद्यालय और कॉलेज

#### 5. क्रेडिट कार्ड लेनदेन

क्रेडिट कार्ड और अन्य सभी लेनदेन की खरीद के लिए केवल डीबीएमएस द्वारा ही संभव किया जाता है। एक क्रेडिट कार्ड धारक अपनी जानकारी के महत्व को जानता है कि सभी डीबीएमएस के माध्यम से सुरक्षित हैं।



चित्र 8.28 क्रेडिट कार्ड लेनदेन

#### 6. सोशल मीडिया साइट

हम सभी अपने विचार साझा करने और अपने दोस्तों से जुड़ने के लिए सोशल मीडिया वेबसाइटों पर हैं। फेसबुक, ट्विटर, पिंटेरेस्ट और गूगल प्लस जैसे इन सोशल मीडिया अकाउंट्स के लिए रोजाना लाखों यूजर्स साइन अप करते हैं। उपयोगकर्ताओं की सभी जानकारी कैसे संग्रहीत की जाती है और अन्य लोगों से कैसे जुड़ सकते हैं, हाँ यह सब DBMS के कारण संभव है।



चित्र 8.29 सोशल मीडिया साइट

### 7. दूरसंचार

कोई भी टेलीकम्युनिकेशन कंपनी बिना DBMS के अपने बिजनेस के बारे में सोच भी नहीं सकती. इन कंपनियों के लिए कॉल विवरण और मासिक पोस्टपेड बिलों को स्टोर करने के लिए डीबीएमएस जरूरी है।



चित्र 8.30 दूरसंचार

### 8. वित्त

वे दिन बहुत दूर चले गए जब पैसे से जुड़ी जानकारी रजिस्ट्रों और फाइलों में जमा हो जाती थी। आज समय बदल गया है क्योंकि वित्त के साथ बहुत सी चीजें हैं जैसे बिक्री का भंडारण, जानकारी रखना और वित्तीय विवरण प्रबंधन आदि।



चित्र 8.31 वित्त

### 9. सैन्य (मिलिट्री)

सेना लाखों सैनिकों का रिकॉर्ड रखती है और उसके पास बड़ी मात्रा में फाइलें होती हैं जिन्हें सुरक्षित और सुरक्षित रखा जाना चाहिए। चूंकि DBMS सैन्य सूचनाओं को एक बड़ा सुरक्षा आश्वासन प्रदान करता है, इसलिए इसका व्यापक रूप से सेनाओं में उपयोग किया जाता है। डीबीएमएस की मदद से कोई भी व्यक्ति सेकंडों में किसी के बारे में सभी जानकारी आसानी से खोज सकता है।



चित्र 8.32 मिलिट्री

### 10 ऑनलाइन खरीदारी

ऑनलाइन शॉपिंग इन दिनों एक बड़ा चलन बन गया है। कोई भी दुकानों पर जाकर अपना समय बर्बाद नहीं करना चाहता। हर कोई घर से खरीदारी करना चाहता है। तो सभी उत्पादों को DBMS की मदद से ही जोड़ा और बेचा जाता है। खरीदारी की जानकारी, चालान बिल और भुगतान, ये सभी DBMS की मदद से किए जाते हैं।



चित्र 8.33 ऑनलाइन खरीदारी

### 11. मानव संसाधन प्रबंधन

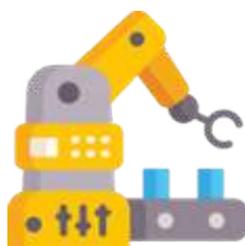
बड़ी फर्मों के नीचे कई कर्मचारी काम करते हैं। मानव संसाधन प्रबंधन विभाग DBMS के माध्यम से प्रत्येक कर्मचारी के वेतन, कर और कार्य का रिकॉर्ड रखता है।



चित्र 8.34 मानव संसाधन प्रबंधन

### 12. उत्पादन

विनिर्माण कंपनियां उत्पाद बनाती हैं और उन्हें दैनिक आधार पर बेचती हैं। मात्रा, बिल, खरीद और आपूर्ति श्रृंखला प्रबंधन जैसे उत्पादों के बारे में सभी विवरणों का रिकॉर्ड रखने के लिए, DBMS का उपयोग किया जाता है।



चित्र 8.35 उत्पादन

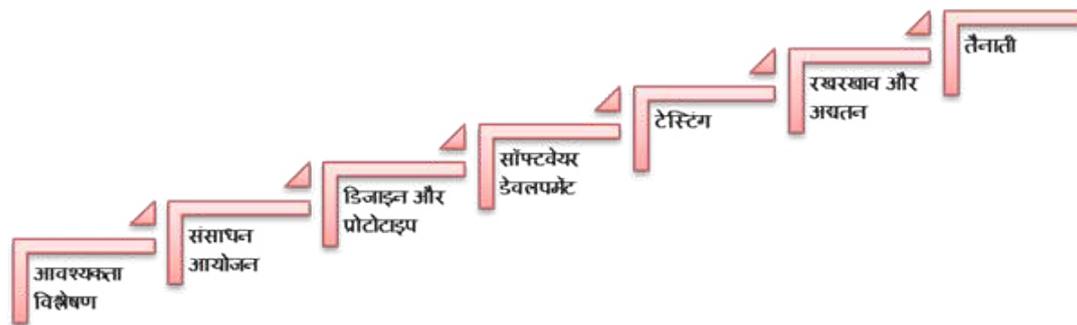
### 13. एयरलाइन आरक्षण प्रणाली

रेलवे आरक्षण प्रणाली के समान, एयरलाइन को उड़ान आगमन, प्रस्थान और देरी की स्थिति का रिकॉर्ड रखने के लिए भी DBMS की आवश्यकता होती है। तो संक्षेप में, कोई कह सकता है कि DBMS का उपयोग हमारे चारों ओर हर जगह किया जाता है और हम DBMS के बिना इस पर भरोसा नहीं कर सकते।



चित्र 8.36 एयरलाइन आरक्षण प्रणाली

### 8.2.6 सॉफ्टवेयर डेवलपमेंट प्रक्रिया



चित्र 8.37 सॉफ्टवेयर डेवलपमेंट प्रक्रिया

## सारांश



- स्टेपवाइज रिफाइनमेंट प्रोग्राम रिफाइनमेंट के लिए एक टॉप-डाउन दृष्टिकोण है जिसमें एक प्रोग्राम को विस्तार के बढ़ते स्तरों के पदानुक्रम में परिष्कृत किया जाता है।
- ऑनलाइन शॉपिंग इन दिनों एक बड़ा चलन बन गया है। कोई भी दुकानों पर जाकर अपना समय बर्बाद नहीं करना चाहता। हर कोई घर से खरीदारी करना चाहता है।
- कोई भी टेलिकम्युनिकेशन कंपनी बिना DBMS के अपने बिजनेस के बारे में सोच भी नहीं सकती। इन कंपनियों के लिए कॉल विवरण और मासिक पोस्टपेड बिलों को स्टोर करने के लिए डीबीएमएस जरूरी है।
- विनिर्माण कंपनियां उत्पाद बनाती हैं और उन्हें दैनिक आधार पर बेचती हैं। मात्रा, बिल, खरीद और आपूर्ति श्रृंखला प्रबंधन जैसे उत्पादों के बारे में सभी विवरणों का रिकॉर्ड रखने के लिए, DBMS का उपयोग किया जाता है।
- क्रेडिट कार्ड और अन्य सभी लेनदेन की खरीद के लिए केवल डीबीएमएस द्वारा ही संभव किया जाता है। एक क्रेडिट कार्ड धारक अपनी जानकारी के महत्व को जानता है कि सभी डीबीएमएस के माध्यम से सुरक्षित हैं।
- टिकट बुकिंग, ट्रेन प्रस्थान और आगमन की स्थिति का रिकॉर्ड रखने के लिए डेटाबेस की आवश्यकता होती है। इसके अलावा अगर ट्रेनें लेट हो जाती हैं तो लोगों को डेटाबेस अपडेट के जरिए इसका पता चल जाता है।
- डेटाबेस प्रबंधन प्रणाली (डीबीएमएस के रूप में भी जाना जाता है) उपयुक्त सुरक्षा उपायों पर विचार करके उपयोगकर्ताओं के डेटा को संग्रहीत करने और पुनर्प्राप्त करने के लिए सॉफ्टवेयर है। यह उपयोगकर्ताओं को उनकी आवश्यकताओं के अनुसार अपने डेटाबेस बनाने की अनुमति देता है।
- एक आरक्षण प्रणाली का उपयोग मुख्य रूप से सूचनाओं को संग्रहीत करने और पुनः प्राप्त करने के साथ-साथ हवाई यात्रा, कार किराए पर लेने, होटल और अन्य गतिविधियों से संबंधित लेनदेन करने के लिए किया जाता है।
- DBMS किसी एप्लिकेशन से डेटा के लिए अनुरोध स्वीकार करता है और ऑपरेटिंग सिस्टम को विशिष्ट डेटा प्रदान करने का निर्देश देता है। बड़ी प्रणालियों में, एक DBMS उपयोगकर्ताओं और अन्य तृतीय-पक्ष सॉफ्टवेयर को डेटा संग्रहीत करने और पुनर्प्राप्त करने में मदद करता है।
- शिक्षा और मनोरंजन सॉफ्टवेयर शैक्षिक संगठनों के लिए एक शक्तिशाली उपकरण है, विशेष रूप से वे जो छोटे बच्चों के साथ काम करते हैं। कंप्यूटर गेम, शैक्षिक खेल, अनुवाद सॉफ्टवेयर, मैपिंग सॉफ्टवेयर और मनोरंजन सॉफ्टवेयर के अन्य रूप उपलब्ध हैं।

## अभ्यास

निम्नलिखित प्रश्नों के उत्तर दो:

डीबीएमएस को परिभाषित कीजिए।

सॉफ्टवेयर विकास प्रक्रिया को समझाइए।

डीबीएमएस के अनुप्रयोग और उपयोग क्या हैं?

डीबीएमएस की विशेषताएं क्या हैं?

रिक्त स्थानों की पूर्ति करें

\_\_\_\_\_ सॉफ्टवेयर शैक्षिक संगठनों के लिए एक शक्तिशाली उपकरण है, विशेष रूप से वे जो छोटे बच्चों के साथ काम करते हैं।

\_\_\_\_\_ किसी एप्लिकेशन से डेटा के लिए अनुरोध स्वीकार करता है और ऑपरेटिंग सिस्टम को विशिष्ट डेटा प्रदान करने का निर्देश देता है।

विनिर्माण कंपनियां उत्पाद बनाती हैं और उन्हें दैनिक आधार पर बेचती हैं। उत्पादों के बारे में सभी विवरणों का रिकॉर्ड रखने के लिए जैसे \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, और \_\_\_\_\_, DBMS का उपयोग किया जाता है।

सेना लाखों सैनिकों का रिकॉर्ड रखती है और उसके पास बड़ी मात्रा में फाइलें होती हैं जिन्हें \_\_\_\_\_ रखा जाना चाहिए।

\_\_\_\_\_ सॉफ्टवेयर की सबसे अधिक उपयोग की जाने वाली श्रेणी है और इसका उपयोग व्यावसायिक अनुप्रयोगों का समर्थन करने के लिए किया जाता है।





## 9. समावेशी और पर्यावरण की दृष्टि से टिकाऊ कार्यस्थल



IT - ITeS SSC  
NASSCOM

यूनिट 9.1: सतत अभ्यास

यूनिट 9.2: विविधता का सम्मान करें और समानता को बढ़ावा देने के लिए प्रथाओं को मजबूत करें



## मुख्य सीख

इस मॉड्यूल के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. ऊर्जा दक्षता और अपशिष्ट प्रबंधन के लिए कार्यस्थल में स्थायी प्रथाओं का वर्णन करें
2. लैंगिक समानता बनाए रखने और PwD के लिए समावेशिता बढ़ाने के लिए विभिन्न दृष्टिकोण लागू करें

## यूनिट 9.1: सतत अभ्यास

### यूनिट के उद्देश्य

इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. विभिन्न कार्यों में बिजली/ऊर्जा, सामग्री और पानी के उपयोग को अनुकूलित करने का तरीका प्रदर्शित करें
2. ऊर्जा दक्ष प्रणालियों के कार्यान्वयन की प्रक्रिया को चरणबद्ध तरीके से समझाएं
3. निपटान या कुशल अपशिष्ट प्रबंधन के लिए उत्पन्न पुनर्चक्रण योग्य, गैर-पुनर्नवीनीकरण और खतरनाक कचरे की पहचान करना और उन्हें अलग करना

### 9.1.1 बिजली/ऊर्जा, सामग्री और पानी के उपयोग का अनुकूलन करें

कार्यालय परिसर के भीतर और आसपास और अन्य कॉर्पोरेट वातावरण में हरियाली न केवल कार्यस्थल की सजावट को बढ़ाने में मदद करती है, बल्कि कर्मचारियों की उत्पादकता पर भी सकारात्मक प्रभाव डालती है। हरियाली लोगों को काम पर ध्यान केंद्रित करने में मदद करती है, श्रमिकों और आगंतुकों के बीच सकारात्मक ऊर्जा पैदा करती है।

हरियाली की शुरुआत के अलावा, ऊर्जा का संरक्षण और उपयोग का अनुकूलन भी उतना ही महत्वपूर्ण है। कुछ आवश्यक उपकरण और उपकरण हैं जो हर कार्यस्थल में उपयोग किए जाते हैं, जिनके लिए बिजली की आवश्यकता होती है। उदाहरण के लिए, एयर कंडीशनर, लाइट, पंखा, कंप्यूटर, कॉफी वैंडिंग मशीन ऐसे विद्युत उपकरण या उपकरण हैं जो कार्यालयों में बड़े पैमाने पर उपयोग किए जाते हैं। इसी तरह, वॉशरूम में स्थिर पानी की आपूर्ति एक और महत्वपूर्ण आवश्यकता है। इन सभी आवश्यक ऊर्जा या वस्तुओं का इष्टतम उपयोग ऊर्जा के संरक्षण और पर्यावरण के अनुकूल कार्य वातावरण बनाने के लिए बिल्कुल महत्वपूर्ण है।

हरियाली क्या करती है?

कार्यस्थलों में पौधे हवा को शुद्ध करते हैं; वे CO<sub>2</sub> (कार्बन डाइऑक्साइड गैस) और अन्य वाष्पशील कार्बनिक यौगिकों की सांद्रता को कम करते हैं, जिससे हवा ताजा और स्वस्थ रहती है

बाहरी वनस्पति गर्मियों में कार्यालय ब्लॉक में और उसके आसपास गर्मी को नियंत्रित करती है, गर्मी के तनाव को कम करती है और एयर कंडीशनिंग की आवश्यकता को कम करती है

हरे रंग की छतें और अग्रभाग इन्सुलेशन या गर्मी की अवशोषण क्षमता, घटते ताप और शीतलन व्यय को बढ़ाते हैं कार्यालय भवनों में और आसपास के पौधे जलवाष्प छोड़ते हैं जो हवा को नम कर देता है, जिससे सिरदर्द कम हो जाता है 'हरा दृश्य' भी केंद्र बढ़ाता है, और तनाव से जल्दी ठीक होने में मदद करता है

हरा-भरा वातावरण लोगों को मध्याह्न-भोजन चलना, स्टाफ को सतर्क और स्वस्थ रखने जैसी गतिविधियों को करने के लिए प्रोत्साहित करता है। लंबे समय तक बैठे रहने से सेहत पर बुरा असर पड़ता है।

ऊर्जा कुशल प्रणालियों के कार्यान्वयन की योजना बनाएं

यहां कुछ सरल ऊर्जा प्रबंधन विचार दिए गए हैं जिन्हें कोई भी कार्य केंद्र में लागू कर सकता है।

प्राकृतिक प्रकाश पर्याप्त होने पर कार्यालयों में कृत्रिम प्रकाश व्यवस्था का प्रयोग न करें

जब भी खिड़कियों से पर्याप्त रोशनी उपलब्ध हो, तो पर्दे खोलें और शेड बढ़ाएं

ऊर्जा की बचत करने वाली फ्लोरोसेंट रोशनी और लैंप का उपयोग करें

खाली पड़े कार्यालय स्थानों या सम्मेलन कक्ष जैसे अप्रयुक्त कमरों में रोशनी और उपकरणों को बंद कर दें। सम्मेलन के दौरान लाइट और एसी/पंखे चालू करें

स्नानघर का पंखा और लाइट बंद कर दें जब भी वे व्यस्त न हों  
कार्यालय उपयोगकर्ताओं को व्यर्थ प्रकाश के बारे में याद दिलाने और शिक्षित करने के लिए प्रकाश संवेदक स्थापित करें  
गणक और अन्य कार्यालय उपकरणों के लिए रिचार्जबल बैटरी का उपयोग करें  
उपयोग नहीं किए जाने वाले कंप्यूटरों को बंद कर दें, और कंप्यूटर के ऊर्जा/पावर प्रबंधन उपकरण (यानी स्लीप मोड, हाइबरनेट मोड, स्क्रीन सेवर) का उपयोग करें।  
रात की सफाई के दौरान प्रकाश का उपयोग कम करें  
अगर हीटिंग और एयर कंडीशनिंग चालू है तो कार्यालय के दरवाजे और खिड़कियां बंद रखें  
कार्यालयों में जब एचवीएसी सिस्टम उपयोग में न हों तो उन्हें बंद कर दें  
सुनिश्चित करें कि तापस्थापी सही ढंग से समायोजित हैं  
उच्च दक्षता वाले कार्यालय उपकरण और उपकरणों की खरीद और उपयोग करें  
कार्यालय ऊर्जा खपत के लिए एक स्व-वयस्क प्रणाली स्थापित करें

प्राकृतिक संसाधनों और ऊर्जा के कुशल उपयोग, प्रदूषण में कमी और रोकथाम की दिशा में पहल

ये कुछ माप हैं जो कार्यस्थल में ऊर्जा के उपयोग को अनुकूलित करने में मदद करते हैं। हालांकि, ऊर्जा और अन्य सामग्रियों के उपयोग को अनुकूलित करने का एक अन्य महत्वपूर्ण पहलू उचित रखरखाव है। ऊर्जा और सामग्री संरक्षण को मापने और बनाए रखने के लिए संगठनों को एक चेकलिस्ट तैयार करनी चाहिए। कार्यस्थलों पर ऊर्जा और सामग्री संरक्षण मॉड्यूल के लिए नमूना जांच-सूची निम्नलिखित है।

श्रेणी	जांच-सूची
ऊर्जा प्रबंधन	ऊर्जा प्रबंधन संगठन की स्थापना, और कर्मचारी शिक्षा
	ऊर्जा संरक्षण लक्ष्य और निवेश बजट स्थापना
	ऊर्जा संरक्षण के कार्यान्वयन की स्थिति को समझें
	मासिक उपयोग की माप और रिकॉर्डिंग (बिजली, गैस, तेल और पानी)
	पिछले महीने या वर्ष से अंतर दिखाने वाले ग्राफ सहित आंकड़े तैयार करना
	ऊर्जा तीव्रता की समझ (एमजे/एम2/वर्ष)
	प्रबंधन मानकों की स्थापना
गर्मी स्रोत और गर्मी संदेश देने वाले उपकरण	ठंडा पानी और गर्म पानी के लिए तापमान नियंत्रण
	पंप और पंखे की प्रवाह दर और दबाव का समायोजन
	भाप रिसाव और रोधन प्रबंधन
	दहन उपकरण के वायु अनुपात और निकास गैस का प्रबंधन
	भाप के दबाव और ब्लो-डाउन का नियंत्रण
	ठंडा पानी गुणवत्ता नियंत्रण (विद्युत चालकता)
	मुड़ने वाला फाटक और डैम्पर्स के खुलने का नियंत्रण (जैसे स्वचालित वाल्व)
वातानुकूलन और वेंटिलेशन उपकरण	
	उचित तापमान स्थापना
	उन कमरों के लिए वातानुकूलन बंद करना जो उपयोग में नहीं हैं या खाली नहीं हैं

	<p>उपयुक्त बाहरी वायु सेवन मात्रा का समायोजन</p> <p>संचालन के घंटों की समीक्षा</p> <p>कुल ताप विनिमायक का प्रभावी संचालन (जैसे रोसुनाई)</p> <p>स्थानीय शीतलन और स्थानीय निकास</p> <p>भीतरी वायु गुणवत्ता नियंत्रण (जैसे CO2)</p> <p>हवादार प्रशंसकों के लिए (हाथ-संबंधी या स्वचालित) इन्वर्टर उपकरण की स्थापना</p> <p>4-पाइप वातानुकूलित तंत्रके संचालन में से किसी एक को निलंबित करना, यदि उपयोग किया जाता है</p> <p>गाड़ी अड्डा की जगह में वायु-संचालन का नियंत्रण (सीओ एकाग्रता नियंत्रण)</p>
जल आपूर्ति / जल निकासी और स्वच्छता उपकरण	<p>आपूर्ति किए गए जल प्रवाह और दबाव का नियंत्रण</p> <p>पानी की बचत के उपाय (जैसे पानी की बचत करने वाला टॉप और स्वचालित फ्लशिंग)</p> <p>मौसम के आधार पर ताप स्रोत उपकरण पर तापमान और दबाव स्थापना बदलें</p> <p>गर्म पानी की आपूर्ति परिसंचरण पंप में अंतराल के साथ संचालन</p> <p>वर्षा जल और कुएं के पानी का उपयोग</p> <p>रसोई के उपकरण का प्रबंधन (जैसे खाना पकाने और वाशिंग मशीन)</p>
विद्युत शक्ति प्राप्त करने और सुविधाओं को बदलने का प्रबंधन	<p>मांग का अनुकूलन</p> <p>उपयोग नियंत्रण</p> <p>वोल्टेज समायोजन</p> <p>ऊर्जा घटकप्रबंधन</p>
प्रकाश उपकरणों का संचालन प्रबंधन	<p>इष्टतम रोशनी नियंत्रण</p> <p>जब आवश्यक न हो तो लाइट बंद कर देना (दिन के उजाले का उपयोग)</p> <p>प्रकाश जुड़नार की सफाई और अधिक ऊर्जा-बचत जुड़नार में परिवर्तन</p> <p>गरमागरम लैंप को फ्लोरोसेंट लैंप में बदलें</p> <p>ऊर्जा-बचत एफएफई (फर्नीचर, स्थिरता, और उपकरण) को अपनाना</p>
एलिवेटिंग मशीनों का संचालन और प्रबंधन	<p>संचालन</p> <p>इन्वर्टर नियंत्रण को अपनाना</p> <p>चलती सीढ़ी में मानव गति संवेदकों को अपनाना</p>
इमारतों	<p>खिड़कियों पर सौर विकिरण को रोकना (जैसे छायांकन पर्दे और प्रकाश-परिरक्षण वाली फिल्में)</p> <p>छत पर सौर विकिरण को रोकना (गर्मी परावर्तन कोटिंग)</p>
अन्य	<p>वातानुकूलन और चिलर के लिए संघनक इकाइयों के आसपास जगह बनाए रखें</p> <p>गर्म पानी के झरने से गर्मी का उपयोग</p> <p>ईंधन के रूप में अपशिष्ट पदार्थों का उपयोग करके बॉयलरों की स्थापना</p> <p>सौर ताप का उपयोग</p>

	पवन, सौर और लघु जल विद्युत उत्पादन
	देर रात बिजली का प्रयोग करें
	सह-उत्पादन

तालिका 9.1.1.1: ऊर्जा और सामग्री संरक्षण जांच सूची

अक्षय और गैर-नवीकरणीय सहित विभिन्न ऊर्जा विकल्प

अक्षय ऊर्जा एक शाश्वत ऊर्जा स्रोत है जो दोहन पर समाप्त नहीं होता है और शून्य या न्यूनतम अपशिष्ट उत्पाद प्राप्त करता है। ऊर्जा के ऐसे स्रोत मानव काल के पैमाने पर स्वाभाविक रूप से भर जाते हैं। पर्यावरण और सतत विकास पर पेरिस स्थित स्वायत्त प्राधिकरण, अंतर्राष्ट्रीय ऊर्जा एजेंसी (आईईए) बताती है:

"नवीकरणीय ऊर्जा प्राकृतिक प्रक्रियाओं से प्राप्त होती है जो लगातार भर जाती हैं। इसके विभिन्न रूपों में, यह सीधे सूर्य से प्राप्त होती है, या पृथ्वी के भीतर गहरी गर्मी से उत्पन्न होती है। परिभाषा में शामिल बिजली और गर्मी सौर, पवन, महासागर से उत्पन्न होती है, पनबिजली, बायोमास, भूतापीय संसाधन, जैव ईंधन और अक्षय संसाधनों से प्राप्त हाइड्रोजन।"

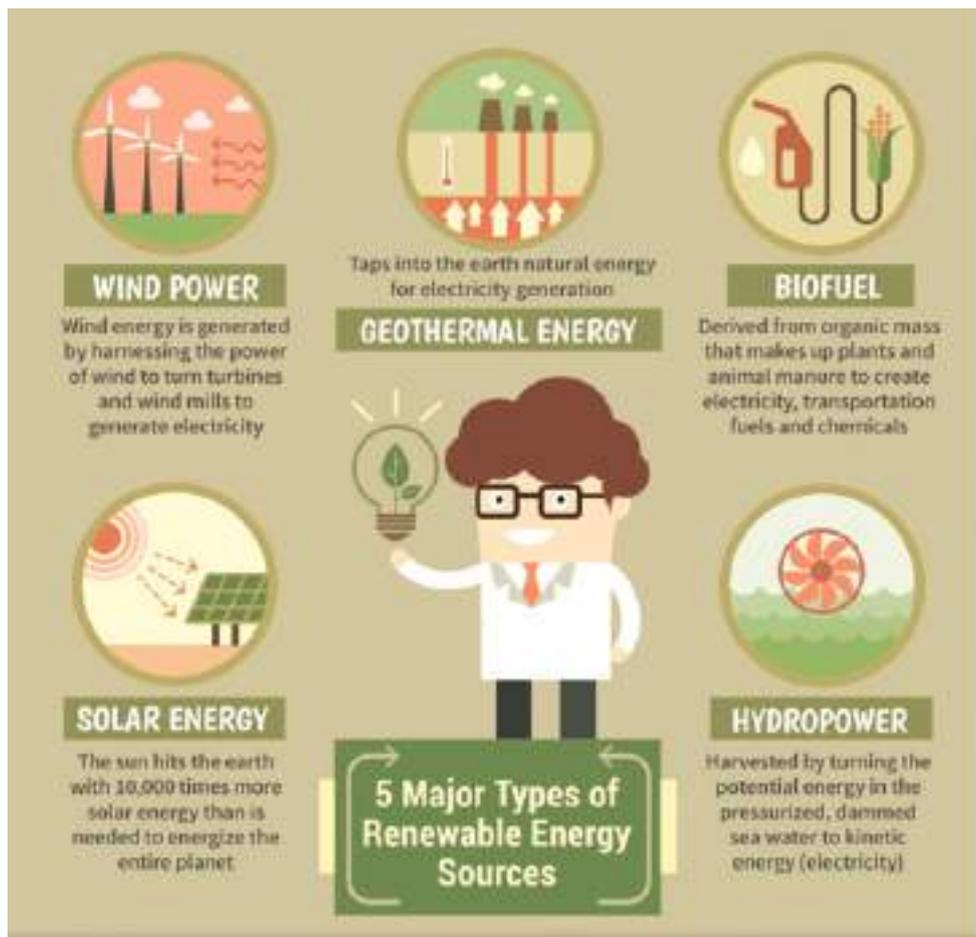
**पवन ऊर्जा:** पवन ऊर्जा सौर ऊर्जा का एक रूप है। पवन ऊर्जा (या पवन ऊर्जा) उस प्रक्रिया का वर्णन करती है जिसके द्वारा बिजली उत्पन्न करने के लिए पवन का उपयोग किया जाता है। पवन टर्बाइन हवा में गतिज ऊर्जा को यांत्रिक शक्ति में परिवर्तित करते हैं। एक जनरेटर यांत्रिक शक्ति को बिजली में परिवर्तित कर सकता है।

**भूतापीय ऊर्जा:** भूतापीय ऊर्जा (ग्रीक जड़ों से भू, अर्थ पृथ्वी, और थर्मस से, जिसका अर्थ है गर्मी) पृथ्वी की पपड़ी के अंदर गर्मी द्वारा बनाई गई ऊर्जा है। यद्यपि सूर्य पृथ्वी की सतह को गर्म करता है, पृथ्वी के अंदर से गर्मी सूर्य के कारण नहीं होती है।

**सौर ऊर्जा:** सौर ऊर्जा सूर्य से प्राप्त ऊर्जा है जो तापीय या विद्युत ऊर्जा में परिवर्तित हो जाती है। सौर ऊर्जा उपलब्ध सबसे स्वच्छ और प्रचुर मात्रा में अक्षय ऊर्जा स्रोत है।

**जैव ऊर्जा:** जैव ऊर्जा प्राकृतिक, जैविक स्रोतों से निर्मित अक्षय ऊर्जा है। आधुनिक तकनीक लैंडफिल या अपशिष्ट क्षेत्रों को संभावित बायोएनेर्जी संसाधन भी बनाती है। इसका उपयोग एक स्थायी शक्ति स्रोत के रूप में किया जा सकता है, जो गर्मी, गैस और ईंधन प्रदान करता है।

**हाइड्रोपावर एनर्जी:** हाइड्रोपावर, या हाइड्रो-एनर्जी, अक्षय ऊर्जा का एक रूप है जो बांधों में जमा पानी का उपयोग करता है, साथ ही साथ नदियों में बहने वाले जलविद्युत संयंत्रों में बिजली पैदा करता है। घूमने वाले ब्लेड एक जनरेटर को घुमाते हैं जो कताई टरबाइन की यांत्रिक ऊर्जा को विद्युत ऊर्जा में परिवर्तित करता है।



चित्र 9.1.1.2: अक्षय ऊर्जा स्रोत

अक्षय ऊर्जा उत्पादन प्रक्रियाओं में ग्रीनहाउस गैसों का उत्सर्जन नहीं करते हैं, जिससे वे पर्यावरणीय क्षरण को रोकने के लिए सबसे स्वच्छ, सबसे व्यवहार्य समाधान बन जाते हैं। कोयले, गैस, तेल और परमाणु जैसे पारंपरिक ऊर्जा स्रोतों की तुलना में - जिनके भंडार सीमित हैं - स्वच्छ ऊर्जा प्राकृतिक चक्रों की उत्पत्ति और अनुकूलन करती है। यह उन्हें एक स्थायी ऊर्जा प्रणाली में एक आवश्यक तत्व बनाता है जो भविष्य की पीढ़ियों को जोखिम में डाले बिना आज विकास की अनुमति देता है।

बिजली प्राथमिक चिकित्सा आपातकालीन प्रक्रियाएं

प्राथमिक चिकित्सा किट में चोटों के लिए प्राथमिक उपचार के लिए बुनियादी उपकरण उपलब्ध होने चाहिए, जिनमें शामिल हैं:

घाव, खरोंच, पंचर, चराई और छींटे

मांसपेशियों में मोच और खिंचाव

छोटे मोटे जखम

विच्छेदन और/या प्रमुख रक्तस्राव घाव

टूटी हुई हड्डियां

आँख की चोटें

झटका

यह सुनिश्चित करने के लिए प्राथमिक चिकित्सा प्रक्रियाओं को विकसित और कार्यान्वित करना चाहिए कि श्रमिकों को उनके कार्यस्थल में प्राथमिक चिकित्सा की स्पष्ट समझ हो। प्रक्रिया को कवर करना चाहिए:

प्राथमिक चिकित्सा किट के प्रकार और वे कहाँ स्थित हैं

प्राथमिक चिकित्सा सुविधाओं का स्थान जैसे प्राथमिक चिकित्सा कक्ष

प्राथमिक चिकित्सा किट और सुविधाओं के लिए कौन जिम्मेदार है और उन्हें कितनी बार जांचा और बनाए रखा जाना चाहिए

प्राथमिक उपचारकर्ताओं के साथ त्वरित आपातकालीन संचार सुनिश्चित करने के लिए उपयुक्त संचार प्रणाली (उपकरण और प्रक्रियाओं सहित) की स्थापना और रखरखाव कैसे करें

प्राथमिक चिकित्सा की आवश्यकता होने पर उपयोग किए जाने वाले संचार उपकरण और प्रणालियाँ (विशेषकर दूरस्थ और पृथक श्रमिकों के लिए)। इन प्रक्रियाओं में संचार उपकरण का पता लगाने के तरीके के बारे में जानकारी होनी चाहिए, जो उपकरण के लिए जिम्मेदार है और इसे कैसे बनाए रखा जाना चाहिए

प्रत्येक प्राथमिक उपचारकर्ता को कार्य क्षेत्र और शिफ्ट आवंटित किए गए हैं। इन प्रक्रियाओं में प्रत्येक प्राथमिक उपचारकर्ता के नाम और संपर्क विवरण शामिल होने चाहिए

प्राथमिक उपचारकर्ताओं को उचित प्रशिक्षण मिले यह सुनिश्चित करने की व्यवस्था

प्राथमिक चिकित्सा के संबंध में श्रमिकों को उचित जानकारी, निर्देश और प्रशिक्षण सुनिश्चित करने की व्यवस्था

जब कोई कर्मचारी किसी प्राथमिक चिकित्सा की जरूरत के बारे में काम शुरू करता है तो जानकारी की तलाश करना, जिसके लिए गंभीर एलर्जी जैसे चिकित्सा आपात स्थिति में विशिष्ट उपचार की आवश्यकता हो सकती है। एक कार्यकर्ता के स्वास्थ्य के बारे में जानकारी को गोपनीय रखा जाना चाहिए और केवल प्राथमिक उपचारकर्ताओं को कार्यकर्ता की सहमति से प्रदान किया जाना चाहिए

कार्यस्थल पर होने वाली चोटों और बीमारियों की रिपोर्ट कैसे करें

रक्त और शरीर के पदार्थों के संपर्क से बचने के लिए अभ्यास

क्या करें जब कोई कर्मचारी या अन्य व्यक्ति काम पर रहने के लिए बहुत अधिक घायल या बीमार हो, उदाहरण के लिए यदि उन्हें चिकित्सा सेवा, घर या कहीं और जहां वे आराम कर सकते हैं और ठीक हो सकते हैं, के लिए परिवहन के साथ सहायता की आवश्यकता है

कार्यस्थल की गंभीर घटना के बाद प्राथमिक उपचारकर्ताओं और कामगारों की सहायता के लिए डीब्रीफिंग या परामर्श सेवाओं तक पहुंच

यहां एक व्यक्ति को बिजली द्वारा प्राणदण्ड से मुक्त करने के लिए कदम उठाए गए हैं।

मुख्य बिजली बंद करें।



उस व्यक्ति को न छुएं जिसे बिजली का झटका लगा हो।



छड़ी, गत्ते, बांस आदि जैसी अचालक वस्तुओं की सहायता से व्यक्ति को विद्युत स्रोत से निकालने का प्रयास करें।



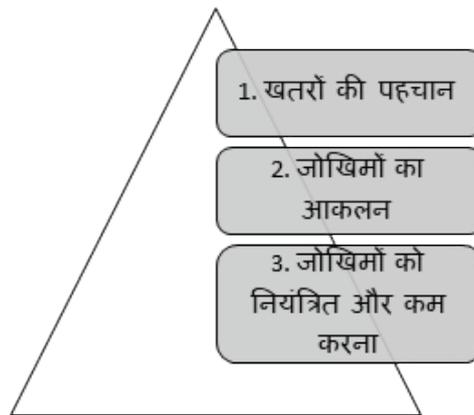
इस स्थिति में व्यक्ति को लेटाओ।



तालिका 9.1.1.3: किसी व्यक्ति को बिजली के झटके से बचाने के उपाय

### 9.1.2 पुनर्चक्रण योग्य, गैर-पुनर्नवीनीकरण और खतरनाक कचरे को अलग करें

खतरे को एक कारक के रूप में परिभाषित किया गया है, जो लोगों और संपत्तियों को समान रूप से नुकसान पहुंचा सकता है, जैसे बिजली, ज्वलनशील उत्पाद, विस्फोटक सामग्री, संक्षारक रसायन, कार्यस्थल पर भारी सीढ़ी का उपयोग करना आदि। सीधे शब्दों में कहें, एक खतरा केवल एक शर्त या परिस्थितियों का एक समूह है। जिससे नुकसान की संभावना बनी रहती है। जोखिम को संभावना या संभावना के रूप में परिभाषित किया गया है कि कोई खतरा वास्तव में किसी को नुकसान पहुंचा सकता है। उदाहरण के लिए, सिगरेट पीने वालों को कैंसर होने का खतरा होता है। संभावित या आसन्न खतरे जो जोखिम और खतरों से संबंधित परिसर को उजागर करते हैं, को खतरे के रूप में जाना जाता है। उदाहरण के लिए, एक व्यक्ति, जो एक इमारत को उड़ाने की क्षमता रखता है, उस इमारत और उसके निवासियों के लिए खतरा है। जोखिम प्रबंधन में शामिल कदम हैं:



चित्र 9.1.2.1: जोखिम प्रबंधन मैट्रिक्स

कार्यस्थल में प्राप्त होने वाली सबसे आम अपशिष्ट सामग्री को निम्नलिखित में वर्गीकृत किया जा सकता है:

तरल अपशिष्ट

कीचड़, गंदा पानी, जैविक तरल पदार्थ, धोने के बाद अपशिष्ट जल

ठोस अपशिष्ट

औद्योगिक स्लैग, प्लास्टिक कचरा, लकड़ी का कचरा, कागज का कचरा, धातु और कांच

जैविक अपशिष्ट

जैव निम्नीकरणीय खाद्य अपशिष्ट, पशु अपशिष्ट, सब्जी अपशिष्ट, उद्यान अपशिष्ट, पशुओं का सड़ा हुआ मांस लैंडफिल में जमा किया जा सकता है या खाद और बायोगैस में परिवर्तित किया जा सकता है

पुनर्चक्रण योग्य अपशिष्ट

कागज, धातु, लकड़ी, जैविक अपशिष्ट आदि का पुनर्चक्रण किया जा सकता है

उपयुक्त रीसाइक्लिंग बिन में रखा जाना चाहिए और कचरे की प्रकृति के अनुसार इलाज किया जाना चाहिए

उदाहरण के लिए जैविक कचरे को खाद और बायोगैस में बदला जा सकता है

खतरनाक अपशिष्ट

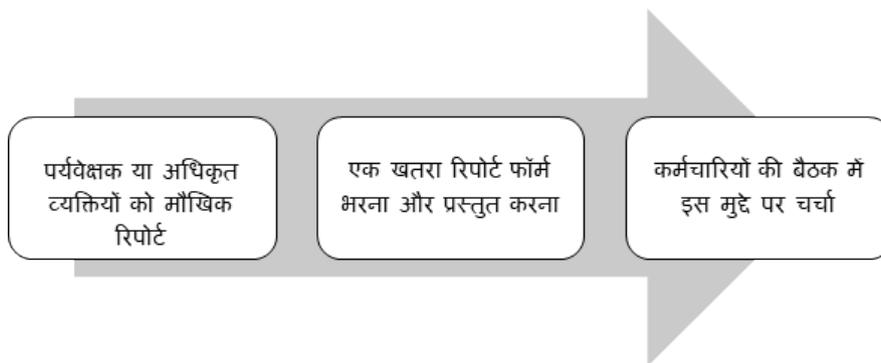
ऐसा कचरा ज्वलनशील, संक्षारक, रेडियोधर्मी, विषाक्त आदि हो सकता है।

ये संभावित रूप से पर्यावरण को नुकसान पहुंचा सकते हैं और उचित उपचार और निपटान के लिए स्पष्ट और स्पष्ट रूप से लेबल वाले डिब्बे में रखा जाना चाहिए



चित्र 9.1.2.2: अपशिष्ट पृथक्करण और निपटान डिब्बे

खतरों और संभावित जोखिमों / खतरों की पहचान की जा सकती है और फिर पर्यवेक्षकों या अन्य अधिकृत व्यक्तियों को निम्नलिखित तरीकों से सूचित किया जा सकता है:



चित्र 9.1.2.3: संभावित खतरों की रिपोर्टिंग का प्रवाह संचित्र

खतरे की पहचान का मतलब है कि काम आधा हो गया है। खतरों के खिलाफ पर्याप्त एहतियाती उपाय करने के लिए, कार्यस्थल में आमतौर पर पाए जाने वाले खतरों की पहचान करने की आवश्यकता है। खतरे की पहचान के सामान्य तरीके हैं:

जॉब हैजर्ड एनालिसिस (JHA)

कर्मचारियों को चोट लगने के जोखिम को कम करने के लिए, नौकरी की भूमिका में विशिष्ट कार्यों से जुड़े खतरों की पहचान करने के लिए यह एक लोकप्रिय तकनीक है।

JHA के सफलतापूर्वक संचालन में शामिल कदम हैं:

संपूर्ण कार्य भूमिका को छोटे-छोटे कार्यों या चरणों में विभाजित करें

आइए एक उदाहरण की मदद से अवधारणा को समझते हैं, जहां कस्टमर केयर एक्जीक्यूटिव जैसे कॉर्पोरेट कार्य पर JHA संचालित किया जा रहा है।

कदम	खतरों से जुड़े	सिफारिशें
व्यापार के उपकरण और उपकरण को संभालना		
कार्यक्षेत्र के सामान्य विद्युत उपकरणों के साथ कार्य करना		
नौकरी की भूमिका का तनाव कारक		

तालिका 9.1.2.4: खतरे की पहचान के लिए झांझा चेकलिस्ट

प्रश्न पूछकर प्रत्येक चरण से जुड़े खतरों का पता लगाएं:

इस कार्य में क्या गलत हो सकता है?

यदि कार्य गलत हो गया तो परिणाम क्या होंगे?

कार्य गलत कैसे हो सकता है?

अन्य योगदान कारक क्या हैं?

क्या संभावनाएं हैं कि यह खतरा होगा?

कर्मचारियों के साथ खतरों के दायरे की समीक्षा करें और चर्चा करें, जो वास्तव में हाथ में काम करेंगे

खतरों को कम करने या उनसे बचने के लिए रणनीतियों और तरीकों का पता लगाएं  
समय-समय पर झाड़ा की समीक्षा और संशोधन करें

खतरा और संचालन क्षमता (HAZOP) अध्ययन

इस तकनीक में मौजूदा पद्धति/प्रक्रिया की एक संरचित और व्यवस्थित परीक्षा शामिल है, इस प्रकार, बदले में, संबंधित खतरों की पहचान करना और उनका आकलन करना।

इन खतरों को प्रक्रिया मापदंडों (भौतिक स्थितियों और प्रवाह, दबाव, तापमान, आर्द्रता, आदि जैसे तत्वों) में विचलन के रूप में आसानी से पहचाना जा सकता है।

विचलन की गंभीरता को विशिष्ट और पूर्व निर्धारित गाइड वर्ड्स की सहायता से स्पष्ट किया जा सकता है।

एक विचलन एक ऐसा तरीका है जिसमें प्रक्रिया की स्थिति अपेक्षित मूल्यों से भटक जाती है।

मार्गदर्शक शब्द + प्रक्रिया की स्थिति = विचलन।

उदाहरण के लिए, **No + Signal = No Signal**

HAZOP के संचालन में शामिल कदम हैं:

संपूर्ण व्यवस्था या प्रक्रिया को अनुभागों या घटकों में अलग करना

एक अध्ययन नोड या बिंदु चुनें

अपेक्षित परिणाम को परिभाषित करें

अपेक्षित परिणाम के आधार पर एक प्रक्रिया पैरामीटर चुनें

एक उपयुक्त गाइड वर्ड लागू करें

विचलन के पीछे का कारण निर्धारित करें

उस कारण से शुरू करें जो सबसे खराब संभावित परिणाम का कारण बन सकता है

इस प्रकार पाए गए विचलन का आकलन करें

कार्रवाई तैयार करें और निर्धारित करें

रिकॉर्ड और दस्तावेज़ जानकारी

B से प्रक्रिया दोहराएं

मार्गदर्शक शब्द और उनके अर्थ के सामान्य उदाहरण हैं:

मार्गदर्शक शब्द	अर्थ	उदाहरण
नहीं (नहीं, कोई नहीं)	वांछित परिणाम में से कोई भी हासिल नहीं हुआ है	संचित गंदगी के कारण गैस कटिंग नोजल के माध्यम से गैस का प्रवाह नहीं होता है
अधिक (इससे अधिक, अधिक)	एक निश्चित प्रक्रिया पैरामीटर में मात्रात्मक वृद्धि	काटने के संचालन के दौरान अपेक्षा से अधिक गर्मी उत्पन्न और उच्च तापमान प्राप्त हुआ

कम (से कम, कम)	एक निश्चित प्रक्रिया प्राचल में मात्रात्मक कमी	अपेक्षा से कम दबाव
साथ ही (इसके अलावा)	सभी बनावट इरादे हासिल किए जाते हैं और एक अतिरिक्त गतिविधि होती है	सभी वाल्व एक ही समय में बंद हो जाते हैं
उल्टा	बनावट इरादे का तार्किक विपरीत होता है	ऊर्जा छेदन यंत्र बिजली आपूर्ति बंद करने के बाद भी ड्रिलिंग जारी रखती है
के अलावा अन्य	एक अप्रत्याशित गतिविधि होती है	गैस सिलेंडर में तरल ईंधन की उपस्थिति

तालिका 9.1.2.5: मार्गदर्शक शब्द और उनकी व्याख्या

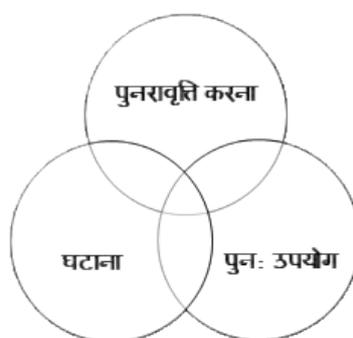
### अपशिष्ट अनुकूलन के 3R

संसाधन अनुकूलन - कचरे माल का पूर्ण उपयोग किया जाना चाहिए, ताकि कचरे माल को तैयार उत्पादों में परिवर्तित करते समय न्यूनतम अपशिष्ट प्राप्त किया जा सके।

स्क्रेप सामग्री का पुनर्चक्रण - जब स्क्रेप बनाया जाता है, तो उसे तुरंत निर्माण प्रक्रिया में शामिल किया जाना चाहिए, ताकि वे कचरे माल के रूप में पूरी तरह से पुनः उपयोग हो सकें।

उन्नत गुणवत्ता नियंत्रण - इसे प्रति बैच अस्वीकृत की संख्या को कम करके कार्यान्वित किया जा सकता है। निरंतर निगरानी के साथ, सावधानीपूर्वक निरीक्षण की उच्च आवृत्ति के साथ यह आसानी से प्राप्त किया जा सकता है।

कचरे का आदान-प्रदान - कुछ कचरे को निर्माण प्रक्रिया से पूरी तरह समाप्त नहीं किया जा सकता है। इस तरह के कचरे को अपशिष्ट विनिमय तकनीकों के माध्यम से प्रभावी ढंग से प्रबंधित किया जा सकता है, जहां एक निश्चित प्रक्रिया में प्राप्त कचरा दूसरे का कचरा माल बन जाता है, और इसके विपरीत।



### चित्र 9.1.2.6: अपशिष्ट अनुकूलन का 3R

#### गड़कों की भराई

अपशिष्ट, जिसे पुनर्नवीनीकरण नहीं किया जा सकता है, जमा किया जाता है और उसके ऊपर मिट्टी की एक परत डाली जाती है

#### भस्मीकरण

कचरे का नियंत्रित दहन शामिल है

90% कचरे की मात्रा कम हो जाती है और राख, गैसों और गर्मी जैसी ज्वलनशील, हल्के वजन वाली सामग्री में परिवर्तित हो जाती है

गैसों को पर्यावरण में छोड़ा जाता है जबकि ताप का उपयोग बिजली उत्पादन में किया जाता है

जैव गैस

जैविक अपशिष्ट जैव निम्नीकरणीय होते हैं और कुछ कवक और शलाकाणु की मदद से जैव गैस संयंत्रों में जैव गैस में परिवर्तित किए जा सकते हैं

जैव गैस के उत्पादन के बाद अवशेष, खाद के रूप में उपयोग किया जाता है

खाद उत्पादन

जैविक कचरे को अक्सर मिट्टी की क्यारियों के नीचे दबा दिया जाता है

वे पोषक तत्वों और खनिजों से भरपूर समृद्ध खाद में विघटित हो जाते हैं

कृमि खाद

कृमियों की मदद से जैविक कचरे को खाद में बदलना शामिल है

कीड़े जैविक कचरे को खाते हैं और उन्हें खाद में बदल देते हैं

## यूनिट 9.2: विविधता का सम्मान करें और समानता को बढ़ावा देने के लिए प्रथाओं को मजबूत करें

### यूनिट के उद्देश्य

इस यूनिट के अंत में, प्रतिभागी निम्न कार्य में सक्षम होंगे:

1. संगठन की विविधता नीति की व्याख्या करें
2. एक अनुकूलनीय और न्यायसंगत कार्य वातावरण के लिए पीडब्ल्यूडी की समावेशी नीतियों का अनुपालन

### 9.2.1 लिंग की अवधारणा, लिंग समानता और लिंग भेदभाव

सहकर्मियों के साथ काम करते समय लैंगिक समावेश, समानता और स्थिरता के बारे में नीतियां और प्रक्रियाएं

भारत का संविधान रोजगार या स्वस्थ या विकलांग से संबंधित मामलों में सभी नागरिकों (भारत के प्रत्येक कानूनी नागरिक, चाहे वे विकलांग हैं) के लिए अवसर की समानता पर समान रूप से लागू होता है। संविधान के तहत राज्य के तहत किसी भी कार्यालय में नियुक्ति। तथ्य की बात के रूप में, एक संगठन के कर्मचारी प्रमुख विविधता का गठन करते हैं। वे विभिन्न क्षेत्रों से आते हैं, विभिन्न सांस्कृतिक और धार्मिक मान्यताओं के साथ। हालांकि, नियोक्ता को लिंग, संस्कृति, धर्म के बावजूद प्रत्येक कर्मचारी को समान अवसर प्रदान करना चाहिए। विशेष रूप से, भारत सरकार ने कार्यस्थल में लैंगिक समानता सुनिश्चित करने के लिए कई उपाय किए हैं। कार्यस्थल पर महिलाओं के अधिकार को स्थापित करने के लिए सरकार ने विधेयक पारित किए हैं। कार्यस्थल पर महिलाओं का यौन उत्पीड़न (रोकथाम, निषेध और निवारण) अधिनियम, 2013 भारत में एक विधायी अधिनियम है जो महिलाओं को उनके कार्यस्थल पर यौन उत्पीड़न से बचाने का प्रयास करता है। इसे 3 सितंबर 2012 को लोकसभा (भारतीय संसद का निचला सदन) द्वारा पारित किया गया था। इसे 26 फरवरी 2013 को राज्य सभा (भारतीय संसद का ऊपरी सदन) द्वारा पारित किया गया था।

नीति की प्रमुख विशेषताओं में शामिल हैं:

अधिनियम कार्यस्थल पर यौन उत्पीड़न को परिभाषित करता है और शिकायतों के निवारण के लिए एक तंत्र बनाता है। यह झूठे या दुर्भावनापूर्ण आरोपों के खिलाफ सुरक्षा उपाय भी प्रदान करता है।

इस अधिनियम में यौन उत्पीड़न के रूप में 'क्विड प्रो क्वो उत्पीड़न' और 'शत्रुतापूर्ण कार्य वातावरण' की अवधारणाओं को भी शामिल किया गया है यदि यह यौन उत्पीड़न के किसी कार्य या व्यवहार के संबंध में होता है।

"पीड़ित महिला" की परिभाषा, जिसे अधिनियम के तहत सुरक्षा मिलेगी, सभी महिलाओं को कवर करने के लिए अत्यंत व्यापक है, चाहे उनकी उम्र या रोजगार की स्थिति कुछ भी हो, चाहे वे संगठित या असंगठित क्षेत्रों में हों, सार्वजनिक हों या निजी और ग्राहकों, ग्राहकों और घरेलू कार्यकर्ता भी।

एक नियोक्ता को ऐसे किसी भी व्यक्ति के रूप में परिभाषित किया गया है जो कार्यस्थल के प्रबंधन, पर्यवेक्षण और नियंत्रण के लिए जिम्मेदार है और इसमें ऐसे व्यक्ति शामिल हैं जो धारा 2(जी) के तहत ऐसे संगठन की नीतियां तैयार और प्रशासित करते हैं।

जबकि विशाखा दिशानिर्देशों में "कार्यस्थल" पारंपरिक कार्यालय सेट-अप तक ही सीमित है जहां स्पष्ट नियोक्ता-कर्मचारी संबंध है, अधिनियम जनता में संगठनों, विभाग, कार्यालय, शाखा इकाई आदि को शामिल करने के लिए बहुत आगे जाता है और निजी क्षेत्र, संगठित और असंगठित, अस्पताल, नर्सिंग होम, शैक्षणिक संस्थान, खेल संस्थान, स्टेडियम, खेल परिसर और परिवहन सहित रोजगार के दौरान कर्मचारी द्वारा दौरा किया गया कोई भी स्थान। यहां तक कि गैर-पारंपरिक कार्यस्थल जिनमें टेली-कन्फ्रेंसिंग शामिल है, इस कानून के तहत शामिल होंगे।

समिति को 90 दिनों की समयावधि में जांच पूरी करनी है। जांच पूरी होने पर, रिपोर्ट नियोक्ता या जिला अधिकारी को भेजी जाएगी, जैसा भी मामला हो, उन्हें 60 दिनों के भीतर रिपोर्ट पर कार्रवाई करने के लिए अनिवार्य है।

प्रत्येक नियोक्ता को प्रत्येक कार्यालय या शाखा में 10 या अधिक कर्मचारियों के साथ एक आंतरिक शिकायत समिति का गठन करना आवश्यक है। जिला अधिकारी को प्रत्येक जिले में और यदि आवश्यक हो तो ब्लॉक स्तर पर एक स्थानीय शिकायत समिति का गठन करना आवश्यक है।

शिकायत समितियों के पास साक्ष्य एकत्र करने के लिए दीवानी न्यायालयों की शक्तियाँ हैं।

यदि शिकायतकर्ता द्वारा अनुरोध किया जाता है, तो शिकायत समितियों को जांच शुरू करने से पहले सुलह का प्रावधान करना होता है।

अधिनियम के तहत जांच प्रक्रिया गोपनीय होनी चाहिए और अधिनियम गोपनीयता भंग करने वाले व्यक्ति पर 5000 रुपये का जुर्माना लगाता है।

अधिनियम में नियोक्ताओं को शिक्षा और संवेदीकरण कार्यक्रम संचालित करने और अन्य दायित्वों के साथ यौन उत्पीड़न के खिलाफ नीतियाँ विकसित करने की आवश्यकता है। जागरूकता निर्माण का उद्देश्य परिसर में प्रदर्शित बैनर और पोस्टर, कर्मचारियों, प्रबंधकों और आंतरिक समिति के सदस्यों के लिए ई-लर्निंग पाठ्यक्रम, कक्षा प्रशिक्षण सत्र, ईमेल के माध्यम से संगठनात्मक यौन उत्पीड़न नीति के संचार, ई-लर्निंग या कक्षा प्रशिक्षण के माध्यम से प्राप्त किया जा सकता है। यह अनुशंसा की जाती है कि ई-लर्निंग या कक्षा प्रशिक्षण कर्मचारी की प्राथमिक संचार भाषा में दिया जाए।

नियोक्ताओं के लिए दंड निर्धारित किया गया है। अधिनियम के प्रावधानों का पालन न करने पर 50,000/- रुपये तक के जुर्माने से दंडित किया जा सकता है। बार-बार उल्लंघन करने पर उच्च दंड और व्यवसाय संचालित करने के लिए लाइसेंस रद्द करना या पंजीकरण रद्द करना पड़ सकता है।

सरकार किसी अधिकारी को किसी भी संगठन में कार्यस्थल और यौन उत्पीड़न से संबंधित रिकॉर्ड का निरीक्षण करने का आदेश दे सकती है।

अधिनियम के तहत, जो स्कूलों और कॉलेजों में छात्रों के साथ-साथ अस्पतालों में रोगियों को भी कवर करता है, नियोक्ताओं और स्थानीय अधिकारियों को सभी शिकायतों की जांच के लिए शिकायत समितियों का गठन करना होगा। अनुपालन करने में विफल रहने वाले नियोक्ताओं को 50,000 रुपये तक के जुर्माने से दंडित किया जाएगा।

## 9.2.2 संगठन के निवारण तंत्र

कार्यस्थल पर लैंगिक समानता के कारण को स्वीकार/मान्य करने, साझा करने और बढ़ावा देने के लिए संचार के समावेशी उपकरण और प्रथाएं

महिलाओं की सुरक्षा और उसके मुद्दों पर दुनिया भर में चर्चा और बहस होती है। फिर भी हर साल यौन उत्पीड़न की रिपोर्ट की संख्या खतरनाक दर से बढ़ रही है। इसलिए, एक निश्चित कंपनी को महिला कर्मचारियों की सुरक्षा सुनिश्चित करने के लिए उनकी अच्छी देखभाल करनी चाहिए।

इसलिए, एक कंपनी को महिलाओं को उन विभिन्न सुविधाओं के बारे में सूचित करना चाहिए जो वे उन्हें प्रदान करने जा रही हैं। कुछ बुनियादी सुविधाओं में निम्नलिखित शामिल हैं।

परिवहन सुविधाएं:

महिला सुरक्षा सुनिश्चित करने में परिवहन एक बड़ी भूमिका निभाता है। यह सुनिश्चित करना कि महिलाओं के साथ भरोसेमंद ड्राइवर होंगे, महिलाओं की सुरक्षा बढ़ाने में मदद करेंगे। रात की यात्राओं के दौरान आप जो सुरक्षा प्रदान कर सकते हैं, उसके बारे में पारदर्शी रहें। प्रत्येक महिला को कंपनी द्वारा प्रदान किए जा सकने वाले विभिन्न सुरक्षा उपायों के बारे में पता होना चाहिए।

दुरुपयोग की रिपोर्ट करना:

जब भी कोई दुर्घटना हो, प्रबंधन को अपने निर्णय लेने में तत्पर होना चाहिए। त्वरित उपचार सुनिश्चित करने के लिए महिला को दुर्व्यवहार की रिपोर्ट करने के तरीके स्पष्ट किए जाने चाहिए।

मातृत्व संबंधी शिकायत:

नियोक्ता को कर्मचारियों के कार्यबल में शामिल होने पर मातृत्व लाभ अधिनियम के तहत उपलब्ध मातृत्व लाभों के बारे में इलेक्ट्रॉनिक रूप से महिलाओं को लिखित रूप में सूचित करना आवश्यक है।

कानून महिला कर्मचारियों को मातृत्व लाभ अवधि के अलावा घर से काम करने की अनुमति देता है यदि काम की प्रकृति इसकी अनुमति देती है।

सीसीटीवी कैमरे

सुनिश्चित करें कि हर स्टेशन सीसीटीवी कैमरों से लैस है जो आजकल यौन उत्पीड़न के मामलों की जांच के लिए सबसे महत्वपूर्ण घटक है।

सुरक्षा प्रहरी

सुनिश्चित करें कि रणनीतिक स्थानों पर पर्याप्त मात्रा में सुरक्षा गार्ड तैनात हैं ताकि महिलाओं की सुरक्षा के लिए किसी भी खतरे को समाप्त किया जा सके। सुनिश्चित करें कि महिलाओं को उन विभिन्न स्थानों के बारे में सूचित किया जाता है जहां सुरक्षा गार्ड मौजूद हैं।

महिला हेल्पलाइन:

महिला हेल्पलाइन नंबर और अन्य महत्वपूर्ण संपर्कों वाला एक पत्रक साझा करें।

जंजीर के ताले / कुंडी

महिलाओं को चेन लॉक और कुंडी प्रदान करें ताकि उनका सामान ठीक से और सुरक्षित रूप से रखा जा सके और किसी भी प्रकार की चोरी से बचा जा सके।

धूम्र संसूचक:

महिलाओं को परिसर के अंदर धूम्र संसूचक के स्थान के बारे में सूचित करें।

इन बुनियादी सुविधाओं को प्रदान करने से यह सुनिश्चित होगा कि महिलाएं बिना किसी डर के आरामदायक आवास का आनंद लें।

सभी प्रकार के लैंगिक भेदभाव, हिंसा और असमानता

कार्यस्थल पर महिलाओं का यौन उत्पीड़न (रोकथाम, निषेध और निवारण) अधिनियम, 2013 कार्यस्थल पर महिलाओं के यौन उत्पीड़न के खिलाफ शिकायतों की जांच और निवारण के लिए एक प्रणाली निर्धारित करता है। यह झूठे या दुर्भावनापूर्ण आरोपों के खिलाफ सुरक्षा उपाय भी प्रदान करता है।

महिलाओं के लिए सुरक्षित कार्य वातावरण सुनिश्चित करने के लिए अधिनियम के प्रमुख प्रावधान नियोक्ताओं के लिए निम्नलिखित जिम्मेदारियां निर्धारित करते हैं:

यौन उत्पीड़न के दंडात्मक परिणाम प्रदर्शित करें

कार्यशालाओं और संवेदीकरण कार्यक्रमों का आयोजन

एक आंतरिक नीति, चार्टर, संकल्प, घोषणा तैयार करना

एक 'आंतरिक शिकायत समिति' (आईसीसी) बनाएं जहां कर्मचारियों की संख्या दस से अधिक हो

समितियों को आवश्यक सुविधाएं प्रदान करें

गवाहों/प्रतिवादी की सुरक्षित उपस्थिति

समिति की रिपोर्ट समय पर प्रस्तुत करने की निगरानी करें

यदि महिला चाहें तो आपराधिक मामले को आगे बढ़ाने में सहायता करें

जांच प्रक्रिया की गोपनीयता बनाए रखें। अधिनियम गोपनीयता भंग करने वाले व्यक्ति पर 5,000 रुपये (US\$68) का जुर्माना लगाता है

यौन उत्पीड़न एक अपराध होने के कारण, नियोक्ता अपराधों की रिपोर्ट करने के लिए बाध्य हैं

कार्यस्थल पर यौन उत्पीड़न की समस्या से निपटने के लिए, कॉर्पोरेट मामलों के मंत्रालय ने 31 जुलाई, 2018 की

अधिसूचना के माध्यम से कंपनी (लेखा) नियम 2014 में संशोधन किया। अधिसूचना निजी कंपनियों के लिए अधिनियम के अनुपालन का खुलासा करना अनिवार्य बनाती है। उनके निदेशकों की वार्षिक रिपोर्ट।

इसके अलावा, अधिनियम स्थानीय शिकायत समिति (एलसीसी) की स्थापना के लिए जिला अधिकारी को सूचित करने के लिए उपयुक्त राज्य सरकार पर जिम्मेदारी डालता है।

जब यौन उत्पीड़न के बारे में सांस्कृतिक दृष्टिकोण बदलने की बात आती है तो मानव संसाधन प्रबंधक अग्रिम पंक्ति में होते हैं।

महिलाओं के लिए सुरक्षित कार्य वातावरण सुनिश्चित करने के लिए मानव संसाधन विकसित कर सकते हैं कि कुछ सर्वोत्तम अभ्यास नीचे दिए गए हैं:

उस आधिकारिक कर्मचारी पुस्तिका को अद्यतन करें जो कार्यस्थल पर यौन उत्पीड़न का अनुभव होने पर होने वाली प्रक्रिया को रेखांकित करती है। एक स्पष्ट बयान शामिल करें कि यौन उत्पीड़न बर्दाश्त नहीं किया जाएगा

एक स्पष्ट, सरल, और आसानी से समझ में आने वाला विवरण दें कि परेशान करने वाला व्यवहार या आचरण क्या होता है, जिसमें कार्यस्थल पर परेशान करने वाले व्यवहार के प्रकारों के उदाहरण भी शामिल हैं।

लैंगिक पहचान और यौन अभिविन्यास पर अधिक ध्यान केंद्रित करने के लिए सभी के लिए प्रशिक्षण लागू करें, और यौन उत्पीड़न का अनुभव करने वालों के संबंध में लिंग तटस्थता पर जोर दें

पुरुष कर्मचारियों को संवेदनशील बनाना और महिलाओं में आगे आने और शिकायत दर्ज करने के लिए विश्वास को मजबूत करना

जहां उनके कर्मचारी रहते हैं या काम करते हैं, वहां रोजगार कानून में बदलाव के बारे में अद्यतन रहें। एचआर को यह सुनिश्चित करने के लिए पेशेवर संघों, कानूनी परामर्शदाताओं और ऑनलाइन संसाधनों का भी उपयोग करना चाहिए कि कंपनी कर्मचारी अधिकारों से संबंधित मौजूदा और आगामी विधायी परिवर्तनों के अनुरूप और जागरूक है।

सहकर्मियों के लिए आंतरिक और बाहरी संचार का उपयोग करें

अक्सर यह कहा जाता है कि किसी का व्यवहार उसके चरित्र का आईना होता है। वास्तव में, आपका व्यवहार बहुत कुछ बताता है कि आप किस तरह के व्यक्ति हैं। यदि आप एक अच्छे व्यवहार वाले व्यक्ति नहीं हैं तो आपकी शैक्षणिक डिग्री का कोई महत्व नहीं है। आपको लगभग हर स्थिति में अच्छा आचरण करने की आवश्यकता है चाहे आप नौकरी के लिए साक्षात्कार के लिए उपस्थित हों या स्नातकोत्तर डिग्री प्राप्त करें, अपने कार्यस्थल पर या अपने ग्राहकों के साथ व्यवहार करते समय, अपने स्कूल / कॉलेज में या पार्टियों में भाग लेते समय। अपने घर पर भी अपने रिश्तेदारों के सामने, यह आपका अच्छा व्यवहार है जो सबसे ज्यादा मायने रखता है। लेकिन व्यवहारिक शिष्टाचार एक ऐसी चीज है जिसे किसी पर थोपा नहीं जा सकता, इसे अपने भीतर ही विकसित और पोषित करना होता है।

महिलाओं के प्रति आज्ञाकारी व्यवहार शिष्टाचार दिखाना बहुत महत्वपूर्ण है।

ऐसे कौन से विभिन्न उदाहरण हैं जहां कोई ऐसा शिष्टाचार दिखा सकता है? चलो एक नज़र डालते हैं:

कमरे में प्रवेश करने से पहले: प्रवेश करने से पहले आपको हमेशा दस्तक देनी चाहिए और अनुमति मांगनी चाहिए। यह शायद सबसे बुनियादी शिष्टाचार है। आपको यह सुनिश्चित करना चाहिए कि महिला की निजता को कोई नुकसान न पहुंचे। तो, दस्तक दें और कमरे में प्रवेश करने से पहले मौखिक अनुमति लें

स्पर्श संपर्क से बचना: आपको हमेशा यह सुनिश्चित करना चाहिए कि आप ग्राहक के व्यक्तिगत स्थान में घुसपैठ न करें। यह न केवल अव्यवसायिक है, बल्कि अस्वच्छ भी है। इसलिए स्पर्श संपर्क से बचने की पूरी कोशिश करें। यदि आवश्यक हो, तो अनुमति मांगें और फिर ग्राहक की सहायता करें।

गाली-गलौज या इशारों का इस्तेमाल करना: यह आखिरी चीज है जिसकी एक महिला/ग्राहक आपसे उम्मीद करता है। सुनिश्चित करें कि आप ग्राहक के सामने कभी भी किसी भी तरह की अभद्र भाषा का प्रयोग न करें। सुनिश्चित करें कि आप मेहमानों के सामने अपने सहयोगियों को गाली न दें

महिलाओं को समाज के साथ-साथ कानूनी रूप से भी सशक्त किया जाता है, बस कुछ बुनियादी अधिकारों की बात करें जो सार्वभौमिक रूप से दोनों लिंगों पर लागू होते हैं लेकिन विशेष रूप से महिलाओं के लिए निम्नलिखित शामिल हैं: -

एक महिला के रूप में सम्मान और सम्मान के अधिकार जिसका अर्थ है कि किसी भी उम्र के किसी भी पुरुष को किसी महिला को छेड़ने या उसका यौन उत्पीड़न करने का अधिकार नहीं है।

सभी परिस्थितियों में सम्मान का विशेषाधिकार: किसी को भी यह अधिकार नहीं है कि वह कार्यस्थल पर, घर पर या सड़कों पर, स्कूल, कॉलेज या किसी सामाजिक सभा में महिलाओं को असहज कर सके।

शारीरिक और मानसिक सुरक्षा का अधिकार: किसी को भी शारीरिक बल प्रयोग करने, शारीरिक या मानसिक रूप से प्रताड़ित करने या किसी भी तरह से महिलाओं को मजबूर करने का अधिकार नहीं है, चाहे उस व्यक्ति के साथ कोई भी संबंध क्यों न हो

शिकायत करने का विशेषाधिकार: महिलाओं को छोटे से छोटे तरीके से भी उल्लंघन किए जाने पर शिकायत करने का पूरा अधिकार है। सलाह लें और ऐसी परिस्थितियों में सही रास्ता अपनाएं, व्यक्ति की स्थिति चाहे जो भी हो, चाहे वह बॉस हो, रिश्तेदार हो या पड़ोस का धमकाने वाला हो

कार्यस्थल पर यौन उत्पीड़न की रोकथाम के लिए विशाखा दिशानिर्देशों के अनुसार संगठन द्वारा सुरक्षा के लिए महिला कर्मचारी के रूप में अधिकार

महिलाओं के खिलाफ हिंसा, शारीरिक या मानसिक उसकी नियति नहीं है जैसा कि कुछ मामलों में बताया जाता है। प्रभावशाली व्यवहार किसी का अधिकार नहीं है और न ही महिलाओं की नियति, इस प्रकार उसके खिलाफ शिकायत करना सही क्रम में है

एक सुरक्षा प्रक्रिया आवश्यक गतिविधियों का एक सेट अनुक्रम है जो एक विशिष्ट सुरक्षा कार्य या कार्य करता है। प्रक्रियाओं को आम तौर पर एक अंतिम परिणाम प्राप्त करने के लिए एक सुसंगत और दोहराव वाले दृष्टिकोण या चक्र के रूप में पालन किए जाने वाले चरणों की एक श्रृंखला के रूप में डिज़ाइन किया गया है।

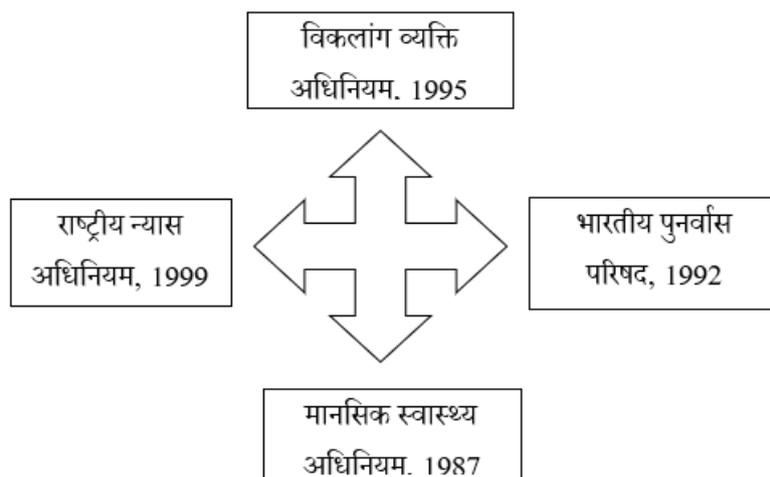
एक बार लागू होने के बाद, सुरक्षा प्रक्रियाएं संगठन के सुरक्षा मामलों के संचालन के लिए स्थापित कार्यों का एक सेट प्रदान करती हैं, जो प्रशिक्षण, प्रक्रिया ऑडिटिंग और प्रक्रिया में सुधार की सुविधा प्रदान करेगी। प्रक्रियाएं सुरक्षा प्रक्रियाओं में भिन्नता को कम करने के लिए आवश्यक स्थिरता को लागू करने के लिए एक प्रारंभिक बिंदु प्रदान करती हैं, जिससे संगठन के भीतर सुरक्षा का नियंत्रण बढ़ जाता है।

एक नियोक्ता को यह सुनिश्चित करना चाहिए कि कर्मचारी सुरक्षा प्रक्रियाओं और संबंधित वातावरण से खतरे में पड़े बिना हर समय सुरक्षित महसूस करें।

### 9.2.3 पीडब्ल्यूडी समावेशी नीतियों का अनुपालन

किसी भी उत्पीड़न से मुक्त एक अनुकूल कार्य वातावरण कैसे बनाए रखें और प्रदान करें; पीडब्ल्यूडी को सुविधाएं और सुविधाएं

भारत सरकार समानता का सम्मान करती है और इसलिए विकलांगता के आधार पर कोई भेदभाव नहीं किया जाना चाहिए। संविधान विकलांगों सहित नागरिकों को न्याय का अधिकार, विचार, अभिव्यक्ति, विश्वास, आस्था और पूजा की स्वतंत्रता, स्थिति और अवसर की समानता और बंधुत्व को बढ़ावा देने के लिए सुरक्षित करता है। किसी भी विकलांग व्यक्ति को किसी विशेष धर्म या धार्मिक समूह के प्रचार और रखरखाव के लिए किसी भी कर का भुगतान करने के लिए मजबूर नहीं किया जा सकता है। इसे लागू करने के लिए, सरकार ने विकलांगों और उनके समानता के अधिकार की रक्षा के लिए कानून पारित किए हैं। विकलांगों से संबंधित कानून इस प्रकार हैं:



चित्र 9.2.3.1: निःशक्तता से संबंधित अधिनियम

विशेष रूप से डिजाइन की गई भर्ती प्रथाओं, पीडब्ल्यूडी के अनुकूल बुनियादी ढांचे, नौकरी की भूमिका आदि के माध्यम से सुधार करें।

एक 'विकलांग व्यक्ति' की परिभाषा को 2016 के अधिनियम के तहत विस्तृत किया गया है: इसमें विकलांग व्यक्तियों, बेंचमार्क विकलांगता वाले व्यक्तियों और उच्च समर्थन की आवश्यकता वाले विकलांग व्यक्तियों को शामिल किया गया है। यह परिभाषा समावेशी है और 21 प्रकार की अक्षमताओं को 'विशिष्ट अक्षमता' के रूप में वर्गीकृत करती है।

यह अधिनियम सरकारी प्रतिष्ठानों के साथ-साथ निजी प्रतिष्ठानों पर भी लागू होता है। कानून के तहत, निजी प्रतिष्ठान एक कंपनी, फर्म, सहकारी या अन्य समाज, संघों, ट्रस्ट, एजेंसी, संस्था, संगठन, संघ, कारखाने, या सरकार द्वारा निर्दिष्ट ऐसे अन्य प्रतिष्ठान को संदर्भित करते हैं।

अधिनियम में सभी प्रतिष्ठानों को समान अवसर नीति बनाने और प्रकाशित करने की आवश्यकता है। विकलांग व्यक्तियों के खिलाफ सभी प्रकार के भेदभाव निषिद्ध हैं, जब तक कि यह साबित नहीं किया जा सकता कि ऐसा भेदभाव प्रकृति में आनुपातिक है और वैध उद्देश्यों को प्राप्त करने का एक आवश्यक साधन है।

अधिनियम बेंचमार्क विकलांग व्यक्तियों के लिए अतिरिक्त लाभ प्रदान करता है, जैसे कि सरकारी प्रतिष्ठानों में रोजगार रिक्तियां, शिक्षा के अवसर, भूमि आवंटन, और गरीबी उन्मूलन योजनाएं, अन्य।

त्वरित न्याय सुनिश्चित करने के लिए, विकलांग व्यक्तियों के अधिकारों के उल्लंघन से संबंधित मामलों से निपटने के लिए प्रत्येक जिले में विशेष अदालतें स्थापित की जाती हैं। विकलांग व्यक्तियों के अधिकारों के उल्लंघन के लिए जुर्माना 7,750 अमेरिकी डॉलर (500,000 रुपये) का आर्थिक जुर्माना और पांच साल तक की कैद हो सकता है।

पीडब्ल्यूडी के उचित मौखिक/अशाब्दिक संचार, योजनाओं और लाभों के लिए उपयोग और वकालत

यद्यपि अधिनियम के तहत अधिकांश अनुपालन केवल सरकारी प्रतिष्ठानों पर लागू होते हैं, निजी प्रतिष्ठान भी अधिनियम के दायरे में आते हैं और उन्हें निम्नलिखित आवश्यकताओं का पालन करना चाहिए:

स्थापना की वेबसाइट पर या प्रतिष्ठान परिसर के भीतर एक विशिष्ट स्थान पर समान अवसर नीति तैयार करें और प्रकाशित करें। नीति में विकलांग व्यक्तियों को कार्यस्थल पर प्रदान किए जाने वाले लाभों और सुविधाओं का विवरण होना चाहिए। नीति की एक प्रति राज्य आयुक्त के पास भी पंजीकृत होनी चाहिए

20 से अधिक कर्मचारियों वाले प्रतिष्ठानों को विकलांग व्यक्तियों की भर्ती और उनके लिए प्रदान की जाने वाली विशेष सुविधाओं की निगरानी के लिए एक संपर्क अधिकारी नियुक्त करना होगा।

नौकरी रिक्तियों की पहचान करने के लिए प्रतिष्ठानों की आवश्यकता है, जो विकलांग व्यक्तियों के लिए उपयुक्त होगा। सरकार से प्रोत्साहन प्राप्त करने वाले प्रतिष्ठानों के मामले में, न्यूनतम पांच प्रतिशत नौकरी रिक्तियों को विकलांग व्यक्तियों के लिए अनिवार्य रूप से आरक्षित किया जाना चाहिए।

नियोक्ता को कार्यस्थल के भीतर विकलांग व्यक्तियों के खिलाफ नाजायज भेदभाव का निषेध सुनिश्चित करना चाहिए

विकलांग कर्मचारियों की पहुंच बढ़ाने के लिए नियोक्ता को अतिरिक्त सुविधाएं या विशेष लाभ प्रदान करना चाहिए, जैसे कि विशेष अवकाश और प्रशिक्षण कार्यक्रमसभी प्रतिष्ठानों को विकलांग व्यक्तियों के संबंध में सरकार द्वारा जारी किए गए पहुंच मानदंडों के अनुरूप होना चाहिए। अभिगम्यता मानदंड कार्यस्थल के बुनियादी ढांचे और संचार प्रौद्योगिकियों से संबंधित हैं, जो विकलांग व्यक्तियों के लिए सुलभ होना चाहिए

प्रत्येक कवर किए गए प्रतिष्ठान को अपने विकलांग कर्मचारियों के रिकॉर्ड बनाए रखने चाहिए

## सारांश

- कार्यालय परिसर के भीतर और आसपास और अन्य कॉर्पोरेट वातावरण में हरियाली न केवल कार्यस्थल की सजावट को बढ़ाने में मदद करती है, बल्कि कर्मचारियों की उत्पादकता पर भी सकारात्मक प्रभाव डालती है।
- कार्यस्थलों में पौधे हवा को शुद्ध करते हैं; वे CO<sub>2</sub> (कार्बन डाइऑक्साइड गैस) और अन्य वाष्पशील कार्बनिक यौगिकों की सांद्रता को कम करते हैं, जिससे हवा ताजा और स्वस्थ रहती है
- बाहरी वनस्पति गर्मियों में कार्यालय ब्लॉक में और उसके आसपास गर्मी को नियंत्रित करती है, गर्मी के तनाव को कम करती है और एयर कंडीशनिंग की आवश्यकता को कम करती है
- हरे रंग की छतें और अग्रभाग इन्सुलेशन या गर्मी की अवशोषण क्षमता, घटते ताप और शीतलन व्यय को बढ़ाते हैं
- कार्यालय भवनों में और आसपास के पौधे जलवाष्प छोड़ते हैं जो हवा को नम कर देता है, जिससे सिरदर्द कम हो जाता है
- खतरे को एक कारक के रूप में परिभाषित किया गया है, जो लोगों और संपत्तियों को समान रूप से नुकसान पहुंचा सकता है, जैसे बिजली, ज्वलनशील उत्पाद, विस्फोटक सामग्री, संक्षारक रसायन, कार्यस्थल पर भारी सीढ़ी का उपयोग करना आदि।
- खतरों के खिलाफ पर्याप्त एहतियाती उपाय करने के लिए, कार्यस्थल में आमतौर पर पाए जाने वाले खतरों की पहचान करने की आवश्यकता है
- भारत का संविधान रोजगार या स्वस्थ या विकलांग से संबंधित मामलों में सभी नागरिकों (भारत के प्रत्येक कानूनी नागरिक, चाहे वे विकलांग हैं) के लिए अवसर की समानता पर समान रूप से लागू होता है।
- कार्यस्थल पर महिलाओं का यौन उत्पीड़न (रोकथाम, निषेध और निवारण) अधिनियम, 2013 कार्यस्थल पर महिलाओं के यौन उत्पीड़न के खिलाफ शिकायतों की जांच और निवारण के लिए एक प्रणाली निर्धारित करता है।
- 2016 के अधिनियम के तहत 'विकलांग व्यक्ति' की परिभाषा को विस्तृत किया गया है: इसमें विकलांग व्यक्ति, बेंचमार्क विकलांगता वाले व्यक्ति और उच्च समर्थन की आवश्यकता वाले विकलांग व्यक्ति शामिल हैं।

## अभ्यास

निम्नलिखित का मिलान करें:

A B

कार्यस्थल पर महिलाओं का यौन उत्पीड़न (रोकथाम, निषेध और निवारण) अधिनियम 1995

विकलांग व्यक्ति अधिनियम 1992

मानसिक स्वास्थ्य अधिनियम 1999

भारतीय पुनर्वास परिषद 2013

राष्ट्रीय न्यास अधिनियम 1987

दिए गए उत्तरों में से सही उत्तर चुनें:

निम्नलिखित में से कौन सा विकल्प गलत है?

हरियाली गर्मी को अवशोषित करती है और कार्यालय की इमारत को ठंडा रखती है

सरकार द्वारा लागू कानून के अनुसार हरियाली अनिवार्य है

हरियाली उत्पादकता बढ़ाती है

आईईए का अर्थ है -

भारतीय ऊर्जा एजेंसी

भारतीय ऊर्जा प्राधिकरण

अंतर्राष्ट्रीय ऊर्जा एजेंसी

अनुपालन करने में विफल रहने वाले नियोक्ता को निम्नलिखित तक के जुर्माने से दंडित किया जाएगा -

INR 50,000

INR 5,00,000

अभी तक निर्धारित करने के लिए

निम्नलिखित प्रश्नों के उत्तर दें

जोखिम प्रबंधन के बुनियादी कदम क्या हैं?

महिला सुरक्षा के संबंध में संगठन के निवारण तंत्र की प्रमुख विशेषताओं को लिखिए।

अक्षय ऊर्जा के सामान्य स्रोत क्या हैं?

पीडब्ल्यूडी नीतियों के उल्लंघन के संभावित परिणाम क्या हो सकते हैं?

किसी व्यक्ति को बिजली के झटके से बचाने के उपाय लिखिए।

## गतिविधि

### ऊर्जा संरक्षण - एक नमूना जांच सूची तैयार करें और निगरानी करें

- यह गतिविधि "एक नमूना चेकलिस्ट तैयार करें और ऊर्जा उपयोग की निगरानी करें" के रूप में है।
- यह गतिविधि प्रशिक्षुओं को कार्यस्थल में ऊर्जा के अनुकूलन को समझने के लिए लक्षित करती है
- प्रशिक्षक कक्षा को तीन समूहों में विभाजित करेगा
- ट्रेनर केस स्टडी के लिए एक विशेष कमरे में अंतर करेगा
- प्रत्येक समूह को निम्नलिखित कार्य सौंपे जाएंगे:
- केस अध्ययन कक्ष में रोशनी, पंखे और एसी की संख्या गिनें
- उनके उपयोग की अवधि नोट करें
- उचित उपयोग और अपव्यय का आकलन करें
- ऊर्जा के उपयोग को इष्टतम करने के तरीके का मूल्यांकन करने के लिए एक चेकलिस्ट तैयार करें
- टिप्पणियों को प्रस्तुत करने वाला एक दस्तावेज जमा करें
- प्रशिक्षक दस्तावेजों की जांच करेगा और सर्वश्रेष्ठ समूह घोषित करेगा।

## गतिविधि

### अपशिष्ट प्रबंधन

- यह गतिविधि "अपशिष्ट प्रबंधन" के रूप में है
- प्रशिक्षक प्रत्येक प्रशिक्षु को एक नमूना जोखिम मापन जांच सूची तैयार करने के लिए कहेगा (जैसा कि इकाई 7.1.2 में दिखाया गया है)
- प्रशिक्षुओं को भवन की अपशिष्ट प्रबंधन प्रणाली का आकलन करना चाहिए
- उन्हें मौजूदा अपशिष्ट प्रबंधन प्रणाली पर एक दस्तावेज तैयार करना चाहिए और इसे बढ़ाने के लिए प्रणालियों का प्रस्ताव देना चाहिए
- उन्हें विभिन्न प्रकार के कचरे और उनके उपचार के बीच पृथक्करण करने में सक्षम होना चाहिए
- प्रशिक्षुओं द्वारा प्रस्तुत दस्तावेज की योग्यता के आधार पर, प्रशिक्षक सर्वश्रेष्ठ रिपोर्ट की घोषणा करेगा
- सर्वोत्तम रिपोर्ट प्रस्तुत करने वाले प्रशिक्षुओं की कक्षा द्वारा सराहना की जाएगी।





## 10. रोजगार कौशल



IT - ITeS SSC  
NASSCOM



रोजगार कौशल (एम्प्लॉयबिलिटी माँड्यूल) को उन सॉफ्ट स्किल्स के रूप में परिभाषित किया जा सकता है जो नियोक्ता एक संभावित कर्मचारी में देखते हैं। ये कौशल कर्मचारियों को उनकी क्षमता और ग्राहकों की संतुष्टि के लिए अपनी भूमिका निभाने के लिए तैयार करते हैं। उदाहरण के लिए, लिखित और मौखिक माध्यम से स्पष्ट और संक्षिप्त तरीके से आपके मतलब की व्याख्या करने की क्षमता ग्राहक या ग्राहक के साथ बेहतर संबंध बनाने में मदद करती है। इसी तरह, काम खत्म करने की समय सीमा के साथ आने वाले तनाव से निपटना और यह सुनिश्चित करना कि आप समय सीमा को पूरा करते हैं, प्रभावी स्व-प्रबंधन प्रशिक्षण के माध्यम से किया जा सकता है। यह किसी कार्य या लक्ष्य को पूरा करने के लिए विभिन्न विषयों, पृष्ठभूमियों और विशेषज्ञता के अन्य लोगों के साथ अच्छी तरह से काम करके भी किया जा सकता है। आज के डिजिटल युग में, नियोक्ता उम्मीद करते हैं कि कर्मचारियों को इंटरनेट के माध्यम से सहयोगी नेटवर्क में सूचना को पुनः प्राप्त करने, एक्सेस करने, स्टोर करने, उत्पादन करने, प्रस्तुत करने और आदान-प्रदान करने के लिए सूचना और संचार प्रौद्योगिकी के प्राथमिक कार्यों का उपयोग करने में सक्षम होना चाहिए। छात्रों को उद्यमशीलता कौशल विकसित करने की आवश्यकता है, ताकि वे अपना व्यवसाय शुरू करने के लिए आवश्यक ज्ञान और कौशल विकसित कर सकें, इस प्रकार नौकरी चाहने वालों के बजाय नौकरी निर्माता बन सकें। संभावित कर्मचारियों को हरित कौशल विकसित करने की आवश्यकता है, जो व्यवसाय, उद्योग और समुदाय में स्थायी सामाजिक, आर्थिक और पर्यावरणीय परिणामों को विकसित करने और समर्थन देने के लिए कार्यबल में आवश्यक तकनीकी कौशल, ज्ञान, मूल्य और दृष्टिकोण हैं। इस प्रकार, छात्रों से कई तरह के कौशल हासिल करने की उम्मीद की जाती है ताकि आप उस संगठन की कौशल मांगों को पूरा कर सकें जिसके लिए आप काम करेंगे या अपना खुद का व्यवसाय स्थापित करने और चलाने के लिए।

यह अध्याय रोजगार कौशल, संवैधानिक मूल्यों, 21वीं सदी में पेशेवर बनने, डिजिटल, वित्तीय और कानूनी साक्षरता, विविधता और समावेशन, अंग्रेजी और संचार कौशल, ग्राहक सेवा, उद्यमिता और प्रशिक्षुता, और नौकरी और करियर के लिए तैयार होने के बारे में है।

दायरे में निम्नलिखित चीजें शामिल हैं:

- रोजगार कौशल का परिचय
- संवैधानिक मूल्य - नागरिकता
- 21वीं सदी में एक पेशेवर बनना
- बुनियादी अंग्रेजी कौशल
- कैरियर विकास और लक्ष्य निर्धारण
- संप्रेषण कौशल
- विविधता और समावेशन
- वित्तीय और कानूनी साक्षरता
- आवश्यक डिजिटल कौशल
- उद्यमिता
- ग्राहक सेवा
- अप्रेंटिसशिप और नौकरियों के लिए तैयार होना

पुस्तिका देखने के लिए क्यूआर कोड स्कैन करें या संबंधित लिंक पर क्लिक करें



<https://eskillindia.org/NewEmployability>

वीडियो देखने के लिए क्यूआर कोड स्कैन करें या संबंधित लिंक पर क्लिक करें



[https://youtu.be/\\_ksGhNu2\\_4E](https://youtu.be/_ksGhNu2_4E)

संगठन में पालन करने के लिए कार्य  
नैतिकता



[https://youtu.be/Sq\\_GY3qvIO0](https://youtu.be/Sq_GY3qvIO0)

आदर्श कार्य संस्कृति में स्वस्थ टीम  
के जुड़ाव का महत्व



[https://youtu.be/\\_XEqD\\_bPZR8](https://youtu.be/_XEqD_bPZR8)

कार्यक्षेत्र को स्वस्थ, संरक्षित और  
सुरक्षित बनाए रखें



<https://youtu.be/SYfOdCsMQV0>

स्वास्थ्य, सुरक्षा और दुर्घटना  
रिपोर्टिंग प्रक्रियाएं और उसका महत्व



<https://youtu.be/kODxk2TDUB8>

डेटा का विश्लेषण करने के लिए  
सटीक प्रोसेस फ्लो का पालन करें





**Skill India**  
कौशल भारत-कुशल भारत



सत्यमेव जयते  
GOVERNMENT OF INDIA  
MINISTRY OF SKILL DEVELOPMENT  
& ENTREPRENEURSHIP



N · S · D · C  
National  
Skill Development  
Corporation

Transforming the skill landscape



IT - ITeS SSC  
NASSCOM

## 11. अनुलग्नक



क्रमिक	यूनिट	यूनिट संख्या	विषय का नाम	पृष्ठ संख्या	क्यूआर कोड के लिए लिंक	क्यूआर कोड
1	3. सॉफ्टवेयर डेवलपमेंट की प्रक्रिया	इकाई 3.1 सॉफ्टवेयर विकास जीवनचक्र	3.1.3 टॉप-डाउन और बॉटम-अप डिजाइन दृष्टिकोण	48	<a href="https://youtu.be/rdkBnY-KOZw">https://youtu.be/rdkBnY-KOZw</a>	 सॉफ्टवेयर डेवलपमेंट
2	4. अनुप्रयोग (एप्लीकेशन) प्रबंधन की प्रक्रिया	इकाई 4.1 सॉफ्टवेयर आवश्यकताएँ विशिष्टता की गुणवत्ता विशेषताएँ	4.1.1 सॉफ्टवेयर आवश्यकताएँ विशिष्टता और विश्लेषण	101	<a href="https://youtu.be/Zcyvcv0-VAI">https://youtu.be/Zcyvcv0-VAI</a>	 सॉफ्टवेयर रिक्वायरमेंट्स स्पेसिफिकेशन की क्वालिटी विशेषताएँ
3	5. पुनः प्रयोज्य कोड डेवलपमेंट की अवधारणा	इकाई 5.1 सॉफ्टवेयर परीक्षण	5.1.1 आंतरिक या बाह्य उपयोगकर्ता सॉफ्टवेयर परीक्षण के लिए सत्यापन और मान्यकरण	122	<a href="https://youtu.be/WsDTMgMLyVw">https://youtu.be/WsDTMgMLyVw</a>	 कांसेप्ट सॉफ्टवेयर टेस्टिंग
4	6. प्रोग्रामिंग और सॉफ्टवेयर डेवलपमेंट के लिए एल्गोरिथ्म	इकाई 6.1 सॉफ्टवेयर प्रोग्रामिंग और एल्गोरिथ्म विकास	6.1.1 विनिर्देशों के अनुसार सॉफ्टवेयर निर्माण के लिए उपयुक्त मानक	141	<a href="https://youtu.be/SSN6nLIUQ6U">https://youtu.be/SSN6nLIUQ6U</a>	 सॉफ्टवेयर प्रोग्रामिंग और एल्गोरिथ्म डेवलपमेंट
5	7. सॉफ्टवेयर अनुप्रयोगों की डिजाइन	इकाई 7.2 सॉफ्टवेयर विकास प्रक्रिया के तत्व	7.2.1 सॉफ्टवेयर डेवलपमेंट प्रोसेस (प्रक्रिया)	163	<a href="https://youtu.be/V0QjCuq5Hms">https://youtu.be/V0QjCuq5Hms</a>	 सॉफ्टवेयर डेवलपमेंट प्रोसेस के एलिमेंट

क्रमिक	यूनिट	यूनिट संख्या	विषय का नाम	पृष्ठ संख्या	क्यूआर कोड के लिए लिंक	क्यूआर कोड
6	8. नौकरी की भूमिका के लिए कार्य आवश्यकताएँ, उपकरण और सॉफ्टवेयर	इकाई 8.2 अनुप्रयोगों और प्रमुख प्रक्रियाओं को विकसित करने के लिए दृष्टिकोण	8.2.1 सॉफ्टवेयर इंजीनियरिंग के लिए विभिन्न दृष्टिकोण	191	<a href="https://youtu.be/h8YjDUvxzVs">https://youtu.be/h8YjDUvxzVs</a>	 <p>एप्लीकेशन और प्रमुख प्रक्रियाओं को विकसित करने के लिए दृष्टिकोण</p>
7				221	<a href="https://youtu.be/_ksGhNu2_4E">https://youtu.be/_ksGhNu2_4E</a>	 <p>संगठन में पालन करने के लिए कार्य नैतिकता</p>
8	10. रोजगार कौशल	रोजगार कौशल		221	<a href="https://youtu.be/Sq_GY3qvIO0">https://youtu.be/Sq_GY3qvIO0</a>	 <p>आदर्श कार्य संस्कृति में स्वस्थ टीम के जुड़ाव का महत्व</p>
9				221	<a href="https://youtu.be/_XEqD_bPZR8">https://youtu.be/_XEqD_bPZR8</a>	 <p>कार्यक्षेत्र को स्वस्थ, संरक्षित और सुरक्षित बनाए रखें</p>

क्रमिक	यूनिट	यूनिट संख्या	विषय का नाम	पृष्ठ संख्या	क्यूआर कोड के लिए लिंक	क्यूआर कोड
10	10. रोजगार कौशल	रोजगार कौशल		221	<a href="https://youtu.be/SYfOdCsMQV0">https://youtu.be/SYfOdCsMQV0</a>	 <p>स्वास्थ्य, सुरक्षा और दुर्घटना रिपोर्टिंग प्रक्रियाएं और उसका महत्व</p>
11				221	<a href="https://youtu.be/kODxk2TDUB8">https://youtu.be/kODxk2TDUB8</a>	 <p>डेटा का विश्लेषण करने के लिए सटीक प्रोसेस फ्लो का पालन करें</p>





**Skill India**  
कौशल भारत - कुशल भारत



सत्यमेव जयते  
GOVERNMENT OF INDIA  
MINISTRY OF SKILL DEVELOPMENT  
& ENTREPRENEURSHIP



**N.S.D.C.**  
National  
Skill Development  
Corporation  
Transforming the skill landscape



ईबुक तक पहुंचने के लिए इस क्यूआर कोड को  
स्कैन/क्लिक करें



**IT - ITes SSC**  
**NASSCOM**

**IT - ITes Sector Skill Council NASSCOM**

**Address:** Plot No. - 7, 8, 9 & 10 Sector - 126, Noida, Uttar Pradesh - 201303

New Delhi - 110049

**Website:** [www.sscnasscom.com](http://www.sscnasscom.com)

**e-mail:** [ssc@nasscom.com](mailto:ssc@nasscom.com)

**Phone:** 0120 4990111 - 0120 4990172

**Price:** ₹