

System Analysis and Design (SAD)

Module 1: Overview system analysis/design concepts

Understand Data processing system

Data processing system

- Definition
 - Function
 - Importance
-

Definition:

A data processing system is a set of software, hardware, and procedures used to process and manipulate data in order to convert it into useful information. These systems are used to automate repetitive tasks and to extract meaningful insights from large volumes of data.

Function:

The primary function of a data processing system is to transform raw data into meaningful information that can be used for decision-making, reporting, or analysis. This involves a series of operations such as validation, sorting, filtering, transformation, and aggregation of data. The operations performed by data processing systems depend on the specific requirements of the organization or the project, but typically involve one or more of the following:

Data input: This involves collecting and entering data into the system, which can be done manually or through automated processes.

Data storage: This involves storing data in a structured manner, such as in databases or files, so that it can be easily retrieved and processed.

Data processing: This involves transforming the input data into a desired format or structure. This can include sorting, filtering, aggregating, and calculating data.

Data output: This involves presenting the processed data in a meaningful way, such as through reports, dashboards, or visualizations.

Data analysis: This involves examining the processed data to identify patterns, trends, and insights that can be used for decision-making or further analysis.

Importance:

Data processing systems are critical for organizations that rely on data to drive their business operations. The importance of data processing systems can be understood by the following points:

Efficiency: Data processing systems automate repetitive tasks, which reduces the time and effort required to process data manually. This leads to increased efficiency and productivity.

Accuracy: Data processing systems ensure that data is processed accurately and consistently, reducing the risk of errors and inconsistencies.

Timeliness: Data processing systems can process data in real-time or near real-time, enabling organizations to make timely decisions based on the latest data.

Insights: Data processing systems can extract meaningful insights from large volumes of data, which can be used to identify opportunities, mitigate risks, and optimize operations.

Competitive advantage: Data processing systems enable organizations to leverage their data assets to gain a competitive advantage, by improving their products and services, reducing costs, and identifying new revenue streams.

In conclusion, data processing systems are critical tools for organizations that rely on data to drive their business operations. These systems transform raw data into meaningful information that can be used for decision-making, reporting, or analysis. By automating repetitive tasks, ensuring accuracy, providing timely insights, and enabling competitive advantage, data processing systems play a crucial role in modern business operations.

Explain Management Information System

Management information system

- Definition
 - Function
 - Importance
-

Definition:

A Management Information System (MIS) is a computerized system that collects, processes, and reports information to support the decision-making process of an organization's management. The system uses data from various sources, such as internal and external databases, to provide timely and accurate information to decision-makers.

Function:

The primary function of an MIS is to provide relevant and timely information to managers to support their decision-making process. This involves collecting data from various sources, processing it to generate meaningful insights, and presenting the insights in a user-friendly format. The specific functions of an MIS include:

MIS को प्राथमिक कार्य व्यवस्थापकहरूलाई उनीहरूको निर्णय प्रक्रियालाई समर्थन गर्न सान्दर्भिक र समयमै जानकारी प्रदान गर्नु हो। यसमा विभिन्न स्रोतहरूबाट डाटा सङ्कलन, अर्थपूर्ण अन्तर्दृष्टिहरू उत्पन्न गर्न प्रशोधन गर्ने, र प्रयोगकर्ता-अनुकूल ढाँचामा अन्तर्दृष्टिहरू प्रस्तुत गर्ने समावेश छ। MIS को विशिष्ट कार्यहरू समावेश छन्:

Data collection: An MIS collects data from various internal and external sources, such as transactional databases, sensors, and social media.

डाटा सङ्कलन: एक MIS ले विभिन्न आन्तरिक र बाह्य स्रोतहरू, जस्तै लेनदेन डाटाबेस, सेन्सर, र सामाजिक मिडियाबाट डाटा सङ्कलन गर्दछ।

Data processing: The collected data is processed to generate meaningful insights, which can be used to support decision-making. This involves various techniques such as data analysis, data mining, and data visualization.

डाटा प्रशोधन: सङ्कलन डाटा अर्थपूर्ण अन्तर्दृष्टि उत्पन्न गर्न प्रशोधन गरिन्छ, जुन निर्णय लिने समर्थन गर्न प्रयोग गर्न सकिन्छ। यसमा डाटा विश्लेषण, डाटा खनन, र डाटा भिजुअलाइजेशन जस्ता विभिन्न प्रविधिहरू समावेश छन्।

Information dissemination: The processed data is presented to the users in a user-friendly format, such as reports, dashboards, or visualizations. This helps managers to understand the insights and make informed decisions quickly and easily.

सूचना प्रसार: प्रशोधित डाटा प्रयोगकर्ता-अनुकूल ढाँचामा प्रयोगकर्ताहरूलाई प्रस्तुत गरिन्छ, जस्तै रिपोर्टहरू, ड्याशबोर्डहरू, वा दृश्यहरू। यसले प्रबन्धकहरूलाई अन्तर्दृष्टि बुझ्न र सूचित निर्णयहरू छिटो र सजिलै गर्न मद्दत गर्दछ।

Decision support: An MIS provides decision support tools to help managers make informed decisions. This can include tools such as simulations, predictive analytics, and optimization models.

निर्णय समर्थन: एक MIS ले प्रबन्धकहरूलाई सूचित निर्णयहरू गर्न मद्दत गर्न निर्णय समर्थन उपकरणहरू प्रदान गर्दछ। यसले सिमुलेशनहरू, भविष्यवाणी गर्ने विश्लेषणहरू, र अनुकूलन मोडेलहरू जस्ता उपकरणहरू समावेश गर्न सक्छन्।

Importance:

The importance of an MIS can be understood by the following points:

महत्व:

एक MIS को महत्व निम्न बिन्दुहरू द्वारा बुझ्न सकिन्छ:

Strategic planning: An MIS provides the information required for strategic planning by providing insights into market trends, customer behaviour, and competitors' activities.

रणनीतिक योजना: एक MIS ले बजार प्रवृत्ति, ग्राहक व्यवहार, र प्रतिस्पर्धी गतिविधिहरू मा अन्तर्दृष्टि प्रदान गरेर रणनीतिक योजना को लागी आवश्यक जानकारी प्रदान गर्दछ।

Decision-making: An MIS supports decision-making by providing timely and accurate information to managers, which enables them to make informed decisions.

निर्णय लिने: एक MIS ले प्रबन्धकहरूलाई समयमै र सही जानकारी प्रदान गरेर निर्णय लिने समर्थन गर्दछ, जसले तिनीहरूलाई सूचित निर्णयहरू गर्न सक्षम बनाउँछ।

Efficiency: An MIS improves the efficiency of an organization by automating repetitive tasks and providing decision support tools to managers.

दक्षता: एक MIS ले दोहोरिने कार्यहरू स्वचालित गरेर र प्रबन्धकहरूलाई निर्णय समर्थन उपकरणहरू प्रदान गरेर संगठनको दक्षता सुधार गर्दछ।

Performance monitoring: An MIS provides information on the performance of an organization, which enables managers to monitor and control operations effectively.

कार्यसम्पादन अनुगमन: एक MIS ले संगठनको कार्यसम्पादन बारे जानकारी प्रदान गर्दछ, जसले प्रबन्धकहरूलाई प्रभावकारी रूपमा सञ्चालन र नियन्त्रण गर्न सक्षम बनाउँछ।

Competitive advantage: An MIS can provide a competitive advantage by enabling organizations to make informed decisions based on real-time data.

प्रतिस्पर्धात्मक लाभ: एक एमआईएसले संस्थाहरूलाई वास्तविक-समय डाटामा आधारित सूचित निर्णयहरू गर्न सक्षम पारेर प्रतिस्पर्धात्मक लाभ प्रदान गर्न सक्छ।

In summary, a Management Information System is a computerized system that collects, processes, and reports information to support the decision-making process of an organization's management. Its functions include data collection, data processing, information dissemination, and decision support. The importance of an MIS lies in its ability to support strategic planning, decision-making, efficiency, performance monitoring, and competitive advantage.

Describe Data support system

Decision support systems

- Definition
 - Function
 - Importance
-

Definition:

A Decision Support System (DSS) is a computer-based information system that supports decision-making activities of individuals, groups, or organizations. It is designed to help decision-makers make effective decisions by providing access to relevant data and tools to analyse it.

परिभाषा:

डिसिजन सपोर्ट सिस्टम (DSS) कम्प्यूटरमा आधारित सूचना प्रणाली हो जसले व्यक्ति, समूह वा संस्थाको निर्णय लिने गतिविधिहरूलाई समर्थन गर्दछ। यसलाई विश्लेषण गर्न सान्दर्भिक डेटा र उपकरणहरूमा पहुँच प्रदान गरेर निर्णय-निर्माताहरूलाई प्रभावकारी निर्णयहरू गर्न मद्दत गर्न डिजाइन गरिएको हो।

Function:

The primary function of a DSS is to provide decision-makers with the information and tools they need to make effective decisions. This involves collecting and analysing data from various sources, such as databases, spreadsheets, and external sources. The specific functions of a DSS include:

समारोहः

DSS को प्राथमिक कार्य भनेको निर्णयकर्ताहरूलाई प्रभावकारी निर्णयहरू गर्न आवश्यक जानकारी र उपकरणहरू प्रदान गर्नु हो। यसमा डाटाबेस, स्प्रेडसिट, र बाह्य स्रोतहरू जस्ता विभिन्न स्रोतहरूबाट डाटा सङ्कलन र विश्लेषण समावेश छ। DSS को विशिष्ट कार्यहरू समावेश छन्:

Data collection: A DSS collects data from various sources, such as databases, spreadsheets, and external sources.

डाटा सङ्कलन: DSS ले विभिन्न स्रोतहरूबाट डाटा सङ्कलन गर्छ, जस्तै डाटाबेस, स्प्रेडसिट र बाह्य स्रोतहरू।

Data analysis: The collected data is analysed using various techniques, such as data mining, statistical analysis, and artificial intelligence.

डाटा विश्लेषण: डाटा माइनिङ, सांख्यिकीय विश्लेषण, र कृत्रिम बुद्धिमत्ता जस्ता विभिन्न प्रविधिहरू प्रयोग गरी सङ्कलन गरिएको डाटाको विश्लेषण गरिन्छ।

Decision modelling: A DSS provides decision modelling tools, such as what-if analysis, optimization, and simulation, to help decision-makers explore different scenarios and outcomes.

निर्णय मोडेलिङ: एक DSS ले निर्णय मोडेलिङ उपकरणहरू प्रदान गर्दछ, जस्तै के-यदि विश्लेषण, अप्टिमाइजेसन, र सिमुलेशन, निर्णय निर्माताहरूलाई विभिन्न परिदृश्यहरू र परिणामहरू अन्वेषण गर्न मद्दत गर्न।

Information presentation: The analysed data and decision models are presented to the decision-makers in a user-friendly format, such as reports, graphs, and charts.

जानकारी प्रस्तुतीकरण: विश्लेषण गरिएको डाटा र निर्णय मोडेलहरू निर्णयकर्ताहरूलाई प्रयोगकर्ता-अनुकूल ढाँचामा प्रस्तुत गरिन्छ, जस्तै रिपोर्टहरू, ग्राफहरू, र चार्टहरू।

Importance:

The importance of a DSS can be understood by the following points:

महत्वः

DSS को महत्व निम्न बिन्दुहरू द्वारा बुझ्न सकिन्छ:

Improved decision-making: A DSS provides decision-makers with access to relevant and timely information, enabling them to make more informed decisions.

सुधारिएको निर्णय लिने: DSS ले निर्णयकर्ताहरूलाई सान्दर्भिक र समय सान्दर्भिक जानकारीमा पहुँच प्रदान गर्दछ, तिनीहरूलाई थप सूचित निर्णयहरू गर्न सक्षम बनाउँछ।

Increased efficiency: A DSS can automate repetitive tasks, such as data collection and analysis, thereby increasing efficiency and productivity.

बढेको दक्षता: DSS ले दोहोरिने कार्यहरूलाई स्वचालित गर्न सक्छ, जस्तै डेटा सङ्कलन र विश्लेषण, जसले गर्दा दक्षता र उत्पादकता बढ्छ।

Enhanced problem-solving: A DSS provides tools to help decision-makers explore different scenarios and outcomes, thereby enhancing problem-solving capabilities.

परिष्कृत समस्या समाधान: DSS ले निर्णय-निर्माताहरूलाई विभिन्न परिदृश्यहरू र परिणामहरू अन्वेषण गर्न मद्दत गर्न उपकरणहरू प्रदान गर्दछ, जसले गर्दा समस्या समाधान गर्ने क्षमताहरू बढाउँछ।

Competitive advantage: A DSS can provide a competitive advantage by enabling organizations to make faster, more informed decisions based on real-time data.

प्रतिस्पर्धात्मक लाभ: एक DSS ले संस्थाहरूलाई वास्तविक-समय डेटामा आधारित छिटो, अधिक जानकारीपूर्ण निर्णयहरू गर्न सक्षम बनाएर प्रतिस्पर्धात्मक लाभ प्रदान गर्न सक्छ।

In summary, a Decision Support System is a computer-based information system that supports decision-making activities of individuals, groups, or organizations. Its functions include data collection, data analysis, decision modelling, and information presentation. The importance of a DSS lies in its ability to improve decision-making, increase efficiency, enhance problem-solving, and provide a competitive advantage.

Understand artificial intelligence

Artificial intelligence

- Definition
 - Function
 - Importance
-

Definition:

Artificial Intelligence (AI) is a field of computer science and engineering that focuses on the development of intelligent machines that can perform tasks that would typically require human intelligence. AI systems are designed to learn from data and experience, make predictions or recommendations, and take actions based on those predictions.

परिभाषा:

आर्टिफिसियल इन्टेलिजेन्स (AI) कम्प्युटर विज्ञान र इन्जिनियरिङको एउटा क्षेत्र हो जसले बौद्धिक मेसिनहरूको विकासमा केन्द्रित हुन्छ जसले सामान्यतया मानव बुद्धिको आवश्यकता पर्ने कार्यहरू गर्न सक्छ। AI प्रणालीहरू डेटा र अनुभवबाट सिक्न, भविष्यवाणी वा सिफारिसहरू गर्न र ती भविष्यवाणीहरूमा आधारित कार्यहरू गर्न डिजाइन गरिएको हो।

Function:

The primary function of AI is to perform tasks that would typically require human intelligence. This involves developing algorithms and models that can learn from data and experience and use that learning to make predictions or recommendations. The specific functions of AI include:

AI को प्राथमिक कार्य भनेको कार्यहरू प्रदर्शन गर्नु हो जुन सामान्यतया मानव बुद्धिको आवश्यकता पर्दछ। यसले डेटा र अनुभवबाट सिक्न सक्ने एल्गोरिदमहरू र मोडेलहरू विकास गर्न र भविष्यवाणी वा सिफारिसहरू बनाउन त्यो सिकाइ प्रयोग गर्न समावेश गर्दछ। AI को विशिष्ट कार्यहरू समावेश छन्:

Machine learning: AI systems use machine learning algorithms to learn from data and experience, and use that learning to make predictions or recommendations.

मेसिन लर्निङ: AI प्रणालीहरूले डेटा र अनुभवबाट सिक्न मेसिन लर्निङ एल्गोरिदमहरू प्रयोग गर्छन्, र त्यो सिकाइलाई भविष्यवाणी वा सिफारिसहरू बनाउन प्रयोग गर्छन्।

Natural language processing: AI systems use natural language processing algorithms to understand and interpret human language.

प्राकृतिक भाषा प्रशोधन: AI प्रणालीहरूले मानव भाषा बुझ्न र व्याख्या गर्न प्राकृतिक भाषा प्रशोधन एल्गोरिदमहरू प्रयोग गर्छन्।

Computer vision: AI systems use computer vision algorithms to interpret and understand visual data, such as images and videos.

कम्प्युटर भिजन: AI प्रणालीहरूले कम्प्युटर भिजन एल्गोरिदमहरू प्रयोग गर्दछन् भिजुअल डेटा, जस्तै छवि र भिडियोहरू व्याख्या गर्न र बुझ्न।

Robotics: AI systems are used in robotics to enable machines to perform tasks that would typically require human intelligence.

रोबोटिक्स: एआई प्रणालीहरू रोबोटिक्समा मेसिनहरूलाई कार्यहरू गर्न सक्षम पार्न प्रयोग गरिन्छ जुन सामान्यतया मानव बुद्धिको आवश्यकता पर्दछ।

Importance:

The importance of AI can be understood by the following points:

महत्व:

AI को महत्व निम्न बिन्दुहरू द्वारा बुझ्न सकिन्छ:

Automation: AI enables automation of tasks that were previously performed by humans, thereby increasing efficiency and productivity.

स्वचालन: AI ले पहिले मानव द्वारा गरिएको कार्यहरूको स्वचालन सक्षम बनाउँछ, जसले गर्दा दक्षता र उत्पादकता बढ्छ।

Decision-making: AI can provide decision support by analyzing data and making predictions or recommendations, thereby improving decision-making.

निर्णय लिने: AI ले डाटा विश्लेषण गरेर र भविष्यवाणी वा सिफारिसहरू बनाएर निर्णय समर्थन प्रदान गर्न सक्छ, जसले गर्दा निर्णय गर्ने क्षमतामा सुधार हुन्छ।

Personalization: AI can personalize user experiences by analyzing user data and providing personalized recommendations and content.

निजीकरण: AI ले प्रयोगकर्ता डेटा विश्लेषण गरेर र व्यक्तिगत सिफारिसहरू र सामग्री प्रदान गरेर प्रयोगकर्ता अनुभवहरू निजीकृत गर्न सक्छ।

Innovation: AI enables the development of new products and services that were not previously possible, such as self-driving cars and virtual assistants.

नवप्रवर्तन: AI ले नयाँ उत्पादन र सेवाहरूको विकासलाई सक्षम बनाउँछ जुन पहिले सम्भव थिएन, जस्तै सेल्फ-ड्राइभिङ कारहरू र भर्चुअल सहायकहरू।

Competitive advantage: AI can provide a competitive advantage by enabling organizations to make faster and more informed decisions, automate tasks, and innovate.

प्रतिस्पर्धात्मक लाभ: AI ले संस्थाहरूलाई छिटो र अधिक जानकारीपूर्ण निर्णयहरू, स्वचालित कार्यहरू, र नवीनताहरू गर्न सक्षम बनाएर प्रतिस्पर्धात्मक लाभ प्रदान गर्न सक्छ।

In summary, Artificial Intelligence is a field of computer science and engineering that focuses on the development of intelligent machines that can perform tasks that would typically require human intelligence. Its functions include machine learning, natural language processing, computer vision, and robotics. The importance of AI lies in its ability to automate tasks, improve decision-making, personalize user experiences, drive innovation, and provide a competitive advantage.

Identify system analysis and design

System analysis/design identify

- Definition
- Function
- Importance

Definition:

System Analysis and Design is a process of developing information systems that involves examining, analyzing, and designing the system to meet the specific needs and requirements of an organization. It involves understanding the current system, identifying areas of improvement, and designing a new system that meets the needs of the organization.

प्रणाली विश्लेषण र डिजाइन सूचना प्रणालीको विकास गर्ने प्रक्रिया हो जसमा संस्थाको विशिष्ट आवश्यकता र आवश्यकताहरू पूरा गर्न प्रणालीको परीक्षण, विश्लेषण र डिजाइन समावेश हुन्छ। यसले हालको प्रणालीलाई बुझ्ने, सुधारका क्षेत्रहरू पहिचान गर्ने, र संगठनको आवश्यकताहरू पूरा गर्ने नयाँ प्रणालीको डिजाइन समावेश गर्दछ।

Function:

The primary function of System Analysis and Design is to develop information systems that meet the needs and requirements of an organization. This involves understanding the current system, identifying areas of improvement, and designing a new system. The specific functions of System Analysis and Design include:

प्रणाली विश्लेषण र डिजाइनको प्राथमिक कार्य भनेको संगठनको आवश्यकता र आवश्यकताहरू पूरा गर्ने सूचना प्रणालीहरू विकास गर्नु हो। यसमा हालको प्रणालीलाई बुझ्ने, सुधारका क्षेत्रहरू पहिचान गर्ने र नयाँ प्रणालीको डिजाइन गर्ने समावेश छ। प्रणाली विश्लेषण र डिजाइन को विशिष्ट कार्यहरू समावेश:

Requirement gathering: System Analysis and Design involves gathering requirements from stakeholders to understand the needs of the organization.

आवश्यकता भेला: प्रणाली विश्लेषण र डिजाइनले संगठनको आवश्यकताहरू बुझ्न सरोकारवालाहरूसँग आवश्यकताहरू भेला गर्ने समावेश गर्दछ।

System analysis: The current system is analyzed to identify areas of improvement, inefficiencies, and potential areas of automation.

प्रणाली विश्लेषण: हालको प्रणालीलाई सुधार, असक्षमता, र स्वचालनका सम्भावित क्षेत्रहरू पहिचान गर्ने विश्लेषण गरिन्छ।

System design: Based on the requirements and analysis, a new system is designed to meet the specific needs and requirements of the organization.

प्रणाली डिजाइन: आवश्यकता र विश्लेषणको आधारमा, संगठनको विशेष आवश्यकता र आवश्यकताहरू पूरा गर्ने नयाँ प्रणाली डिजाइन गरिएको छ।

Implementation: The new system is developed and implemented based on the design.

कार्यान्वयन: नयाँ प्रणाली डिजाइनको आधारमा विकास र कार्यान्वयन गरिन्छ।

Testing: The new system is tested to ensure that it meets the requirements and functions properly.

परीक्षण: नयाँ प्रणालीले आवश्यकताहरू र कार्यहरू ठीकसँग पूरा गर्दछ भनेर सुनिश्चित गर्ने परीक्षण गरिन्छ।

Maintenance: Once the new system is implemented, it requires ongoing maintenance and support to ensure that it continues to function properly.

मर्मतसम्भार: नयाँ प्रणाली लागू भएपछि, यो ठीकसँग काम गर्न जारी छ भनेर सुनिश्चित गर्न निरन्तर मर्मत र समर्थन चाहिन्छ।

Importance:

The importance of System Analysis and Design can be understood by the following points:

प्रणाली विश्लेषण र डिजाइन को महत्व निम्न बिन्दुहरू द्वारा बुझ्न सकिन्छ:

Improved efficiency: System Analysis and Design can identify areas of improvement in the current system and design a new system that is more efficient and effective.

सुधारिएको दक्षता: प्रणाली विश्लेषण र डिजाइनले हालको प्रणालीमा सुधारको क्षेत्रहरू पहिचान गर्न सक्छ र नयाँ प्रणाली डिजाइन गर्न सक्छ जुन अझ प्रभावकारी र प्रभावकारी छ।

Cost savings: System Analysis and Design can identify areas of automation that can reduce costs and improve productivity.

लागत बचत: प्रणाली विश्लेषण र डिजाइनले स्वचालनका क्षेत्रहरू पहिचान गर्न सक्छ जसले लागत घटाउन र उत्पादकता सुधार गर्न सक्छ।

Better decision-making: System Analysis and Design can provide better data and information to support decision-making.

राम्रो निर्णय लिने: प्रणाली विश्लेषण र डिजाइनले निर्णय लिने समर्थन गर्न राम्रो डेटा र जानकारी प्रदान गर्न सक्छ।

Improved customer satisfaction: System Analysis and Design can design a new system that meets the specific needs and requirements of the organization, thereby improving customer satisfaction.

सुधारिएको ग्राहक सन्तुष्टि: प्रणाली विश्लेषण र डिजाइनले संगठनको विशेष आवश्यकता र आवश्यकताहरू पूरा गर्ने नयाँ प्रणाली डिजाइन गर्न सक्छ, जसले गर्दा ग्राहकको सन्तुष्टिमा सुधार हुन्छ।

Competitive advantage: System Analysis and Design can provide a competitive advantage by developing a new system that is more efficient, effective, and meets the specific needs of the organization.

प्रतिस्पर्धात्मक लाभ: प्रणाली विश्लेषण र डिजाइनले नयाँ प्रणाली विकास गरेर प्रतिस्पर्धात्मक लाभ प्रदान गर्न सक्छ जुन अधिक कुशल, प्रभावकारी छ, र संगठनको विशिष्ट आवश्यकताहरू पूरा गर्दछ।

In summary, System Analysis and Design is a process of developing information systems that involves examining, analyzing, and designing the system to meet the specific needs and requirements of an organization. Its functions include requirement gathering, system analysis, system design, implementation, testing, and maintenance. The importance of System Analysis and Design lies in its ability to improve efficiency, reduce costs, support decision-making, improve customer satisfaction, and provide a competitive advantage.

Familiarize with system development and flow

- System analysis
- System design
- Program design
- Programming

- Program test
 - System test
 - Running test
-

System analysis:

System analysis is the process of studying the current system to identify areas of improvement and determine the requirements of the new system. It involves gathering information about the current system, analyzing its strengths and weaknesses, and identifying areas where the new system can improve efficiency and effectiveness.

प्रणाली विश्लेषण सुधारका क्षेत्रहरू पहिचान गर्ने र नयाँ प्रणालीका आवश्यकताहरू निर्धारण गर्ने वर्तमान प्रणालीको अध्ययन गर्ने प्रक्रिया हो। यसमा हालको प्रणालीको बारेमा जानकारी सङ्कलन गर्ने, यसको बलियो र कमजोरीहरूको विश्लेषण गर्ने र नयाँ प्रणालीले दक्षता र प्रभावकारितामा सुधार गर्न सक्ने क्षेत्रहरू पहिचान गर्ने समावेश गर्दछ।

System design:

System design is the process of creating a plan or blueprint for the new system based on the requirements gathered during system analysis. This plan includes system architecture, system components, interfaces, and data structures.

प्रणाली डिजाइन प्रणाली विश्लेषणको क्रममा भेला भएका आवश्यकताहरूको आधारमा नयाँ प्रणालीको लागि योजना वा खाका सिर्जना गर्ने प्रक्रिया हो। यो योजनामा प्रणाली वास्तुकला, प्रणाली कम्पोनेन्टहरू, इन्टरफेसहरू, र डाटा संरचनाहरू समावेश छन्।

Program design:

Program design involves breaking down the system design into smaller components and designing algorithms and data structures for each component. The design must be detailed enough to provide a roadmap for programmers to follow when writing the code.

कार्यक्रम डिजाइनले प्रणाली डिजाइनलाई साना कम्पोनेन्टहरूमा तोड्ने र प्रत्येक कम्पोनेन्टको लागि एल्गोरिदम र डेटा संरचनाहरू डिजाइन गर्ने समावेश गर्दछ। कोड लेख्दा प्रोग्रामरहरूलाई पछ्याउनको लागि रोडम्याप प्रदान गर्ने डिजाइन पर्याप्त विस्तृत हुनुपर्छ।

Programming:

Programming involves writing code in a programming language based on the program design. Programmers use tools like integrated development environments (IDEs) and text editors to write and test the code.

प्रोग्रामिङले प्रोग्राम डिजाइनमा आधारित प्रोग्रामिङ भाषामा कोड लेख्ने समावेश गर्दछ। प्रोग्रामरहरूले कोड लेख्न र परीक्षण गर्न एकीकृत विकास वातावरण (IDEs) र पाठ सम्पादकहरू जस्ता उपकरणहरू प्रयोग गर्छन्।

Program test:

Program testing is the process of testing individual components or modules of the program to ensure that they work correctly. This testing is usually done using automated testing tools or by writing test scripts.

कार्यक्रम परीक्षण भनेको कार्यक्रमको व्यक्तिगत कम्पोनेन्ट वा मोड्युलहरू सही रूपमा काम गरेको सुनिश्चित गर्न परीक्षण गर्ने प्रक्रिया हो। यो परीक्षण सामान्यतया स्वचालित परीक्षण उपकरणहरू प्रयोग गरेर वा परीक्षण लिपिहरू लेखेर गरिन्छ।

System test:

System testing involves testing the entire system to ensure that it meets the requirements specified in the system design. This testing is usually done after the program testing is complete.

प्रणाली परीक्षणले प्रणाली डिजाइनमा निर्दिष्ट आवश्यकताहरू पूरा गर्दछ भनेर सुनिश्चित गर्न सम्पूर्ण प्रणालीको परीक्षण समावेश गर्दछ। यो परीक्षण सामान्यतया कार्यक्रम परीक्षण पूरा भएपछि गरिन्छ।

Running test:

Running tests involve testing the system in a live environment to ensure that it works as expected. This testing is usually done after the system testing is complete and involves testing the system with real users and data.

चलिरहेको परीक्षणहरूले यो अपेक्षित रूपमा काम गर्दछ भनेर सुनिश्चित गर्न प्रत्यक्ष वातावरणमा प्रणालीको परीक्षण समावेश गर्दछ। यो परीक्षण सामान्यतया प्रणाली परीक्षण पूरा भएपछि गरिन्छ र वास्तविक प्रयोगकर्ता र डेटा संग प्रणाली परीक्षण समावेश छ।

Overall, system development and flow involve a series of steps that build upon one another, from system analysis to running tests. Each step is critical to ensure that the final system meets the needs and requirements of the organization.

Prepare major system development models

- Water fall model
- Spiral model
- Others

Waterfall model:

The Waterfall model is a linear, sequential approach to system development. In this model, each phase of the development process must be completed before moving on to the next phase. The phases include requirements gathering and analysis, system design, implementation, testing, and maintenance. The Waterfall model is best suited for projects with well-defined requirements and a predictable development process.

झरना मोडेल प्रणाली विकासको लागि एक रैखिक, क्रमिक दृष्टिकोण हो। यस मोडेलमा, अर्को चरणमा जान अघि विकास प्रक्रियाको प्रत्येक चरण पूरा गर्नुपर्छ। चरणहरूमा आवश्यकताहरू जम्मा गर्ने र विश्लेषण, प्रणाली

डिजाइन, कार्यान्वयन, परीक्षण, र मर्मतसम्भार समावेश छ। झरना मोडेल राम्रो-परिभाषित आवश्यकताहरू र एक अनुमानित विकास प्रक्रिया भएका परियोजनाहरूको लागि सबैभन्दा उपयुक्त छ।

Spiral model:

The Spiral model is a more flexible approach to system development that is based on the Waterfall model. The Spiral model includes four phases: planning, risk analysis, engineering, and evaluation. Each phase is followed by a review process that determines whether to move on to the next phase or return to a previous phase. The Spiral model is best suited for projects with high levels of complexity or uncertainty.

सर्पिल मोडेल प्रणाली विकासको लागि थप लचिलो दृष्टिकोण हो जुन झरना मोडेलमा आधारित छ। सर्पिल मोडेलले चार चरणहरू समावेश गर्दछ: योजना, जोखिम विश्लेषण, इन्जिनियरिङ, र मूल्याङ्कन। प्रत्येक चरण पछि अर्को चरणमा जाने वा अघिल्लो चरणमा फर्कने कि निर्धारण गर्ने समीक्षा प्रक्रियाद्वारा पछ्याइएको छ। सर्पिल मोडेल उच्च स्तरको जटिलता वा अनिश्चितता भएका परियोजनाहरूको लागि उत्तम उपयुक्त छ।

Others: Prototype model,

Agile model:

The Agile model is an iterative and incremental approach to system development that emphasizes collaboration, flexibility, and customer satisfaction. Agile development is based on the principles outlined in the Agile Manifesto, which values individuals and interactions, working software, customer collaboration, and responding to change. Agile development involves working in short iterations or sprints, with a focus on delivering working software quickly and responding to feedback from customers.

एजाइल मोडेल प्रणाली विकासको लागि पुनरावृत्ति र वृद्धिशील दृष्टिकोण हो जसले सहयोग, लचिलोपन, र ग्राहक सन्तुष्टिलाई जोड दिन्छ। फुर्तिलो विकास एजाइल घोषणापत्रमा उल्लिखित सिद्धान्तहरूमा आधारित छ, जसले व्यक्ति र अन्तरक्रिया, काम गर्ने सफ्टवेयर, ग्राहक सहयोग, र परिवर्तनको प्रतिक्रियालाई महत्त्व दिन्छ। फुर्तिलो विकासले छोटो पुनरावृत्ति वा स्प्रिन्टहरूमा काम गर्ने समावेश गर्दछ, काम गर्ने सफ्टवेयरलाई छिटो डेलिभर गर्ने र ग्राहकहरूको प्रतिक्रियालाई प्रतिक्रिया दिनमा केन्द्रित।

Rapid Application Development (RAD) model:

The Rapid Application Development (RAD) model is a fast-paced, iterative approach to system development that emphasizes prototyping and user involvement. The RAD model includes four phases: requirements planning, user design, construction, and cutover. The RAD model is best suited for projects with high user involvement and a need for rapid development.

र्यापिड एप्लिकेसन डेभलपमेन्ट (RAD) मोडेल प्रणाली विकासको लागि द्रुत-गति, पुनरावृत्ति दृष्टिकोण हो जसले प्रोटोटाइपिङ र प्रयोगकर्ता संलग्नतालाई जोड दिन्छ। RAD मोडेलले चार चरणहरू समावेश गर्दछ: आवश्यकताहरू योजना, प्रयोगकर्ता डिजाइन, निर्माण, र कटओभर। RAD मोडेल उच्च प्रयोगकर्ता संलग्नता र द्रुत विकासको आवश्यकता भएका परियोजनाहरूको लागि सबैभन्दा उपयुक्त छ।

DevOps model:

The DevOps model is an approach to system development that emphasizes collaboration and communication between development teams and operations teams. DevOps aims to streamline the development and deployment process by breaking down silos between teams and automating repetitive tasks. The DevOps model is best suited for projects with a focus on continuous delivery and deployment.

DevOps मोडेल प्रणाली विकासको लागि एक दृष्टिकोण हो जसले विकास टोलीहरू र सञ्चालन टोलीहरू बीचको सहकार्य र सञ्चारलाई जोड दिन्छ। DevOps ले टोलीहरू बीच साइलोहरू तोडेर र दोहोरिने कार्यहरूलाई स्वचालित गरेर विकास र तैनाती प्रक्रियालाई सुव्यवस्थित गर्ने लक्ष्य राख्छ। DevOps मोडेल निरन्तर डेलिभरी र तैनातीमा फोकस भएका परियोजनाहरूका लागि सबैभन्दा उपयुक्त छ।

Overall, choosing the right system development model depends on the specific needs and requirements of the project. Each model has its advantages and disadvantages, and it's important to carefully evaluate each option before deciding which one to use.

Module 2: Manage Analysis/Design activities

Identify problem

- Project initiation, Specific sign of problem

During the project initiation phase, it is important to identify potential problems and risks that could impact the success of the project.

परियोजना प्रारम्भ चरणको समयमा, सम्भावित समस्या र जोखिमहरू पहिचान गर्न महत्वपूर्ण छ जसले परियोजनाको सफलतालाई असर गर्न सक्छ।

Some specific signs of problems during project initiation include:

- **Undefined goals and objectives:** If the project goals and objectives are unclear or undefined, it can lead to confusion and misalignment among project stakeholders.
अपरिभाषित लक्ष्य र उद्देश्यहरू: यदि परियोजनाका लक्ष्यहरू र उद्देश्यहरू अस्पष्ट वा अपरिभाषित छन् भने, यसले परियोजना सरोकारवालाहरू बीच भ्रम र गलत तालमेल निम्त्याउन सक्छ।
- **Lack of stakeholder buy-in:** If key stakeholders are not committed to the project, it can lead to resistance and opposition that can delay or derail the project.
सरोकारवाला खरीद-इनको अभाव: यदि मुख्य सरोकारवालाहरू परियोजनाप्रति प्रतिबद्ध छैनन् भने, यसले प्रतिरोध र विरोध निम्त्याउन सक्छ जसले परियोजनालाई ढिलाइ वा पटरीबाट उतार्न सक्छ।
- **Inadequate resource allocation:** If resources such as budget, personnel, and technology are not allocated appropriately, it can lead to delays and compromises in the quality of the project deliverables.
अपर्याप्त स्रोत आवंटन: यदि बजेट, कर्मचारी, र प्रविधि जस्ता स्रोतहरू उचित रूपमा विनियोजन गरिएन भने, यसले परियोजना वितरणको गुणस्तरमा ढिलाइ र सम्झौता निम्त्याउन सक्छ।
- **Unclear project scope:** If the project scope is not clearly defined, it can lead to scope creep, which is the addition of new requirements or features that were not originally planned for.

अस्पष्ट परियोजनाको दायरा: यदि परियोजनाको दायरा स्पष्ट रूपमा परिभाषित गरिएको छैन भने, यसले स्कोप क्रिप निम्त्याउन सक्छ, जुन नयाँ आवश्यकताहरू वा सुविधाहरूको थप हो जुन मूल रूपमा योजना गरिएको थिएन।

- **Lack of project management skills:** If the project team does not have the necessary skills and experience to manage the project, it can lead to ineffective project management and poor decision-making.

परियोजना व्यवस्थापन कौशलको अभाव: यदि परियोजना टोलीसँग परियोजना व्यवस्थापन गर्ने आवश्यक सीप र अनुभव छैन भने, यसले अप्रभावी परियोजना व्यवस्थापन र कमजोर निर्णय लिने निम्त्याउन सक्छ।

Identifying these signs of problems during project initiation is important because it allows project managers to take corrective action early in the project lifecycle. This can help to mitigate risks and ensure that the project is completed successfully within the allotted timeframe and budget.

Identify opportunity for improvement

- Possibilities for improvement, criteria for selection of system projects

Identifying opportunities for improvement involves assessing the current systems and processes used within an organization and identifying areas where improvements can be made. This can be done through a variety of methods, including process mapping, surveys, and benchmarking against industry standards.

Criteria for selecting system projects typically include factors such as:

Alignment with organizational goals: Projects should align with the overall strategic goals of the organization and support its mission and values.

Potential benefits: Projects should provide tangible benefits to the organization, such as increased efficiency, improved customer satisfaction, or increased revenue.

Feasibility: Projects should be technically feasible and should not require excessive resources or time to complete.

Cost-effectiveness: Projects should be cost-effective and should provide a positive return on investment.

Risk management: Projects should be evaluated for potential risks and steps should be taken to mitigate those risks.

Once potential opportunities for improvement have been identified and projects have been selected, a detailed project plan should be developed that includes timelines, resource requirements, and a budget. Effective project management is critical to ensuring that the project is completed successfully and that the expected benefits are realized.

Determine feasibility

- Objectives, Resources

- Types of feasibility; Technical, Economical, Operational
- Making judgement on feasibility

Determining feasibility is a critical step in the system analysis and design process. It involves evaluating the proposed project to determine whether it is technically, economically, and operationally feasible. Here are some details on the different types of feasibility and the process of making a judgment on feasibility:

सम्भाव्यता निर्धारण प्रणाली विश्लेषण र डिजाइन प्रक्रिया मा एक महत्वपूर्ण कदम हो। यसमा प्रस्तावित परियोजना प्राविधिक, आर्थिक र परिचालन रूपमा सम्भव छ कि छैन भनेर निर्धारण गर्न मूल्याङ्कन समावेश छ। सम्भाव्यताका विभिन्न प्रकारहरू र सम्भाव्यतामा निर्णय गर्ने प्रक्रियामा यहाँ केही विवरणहरू छन्:

Objectives: The objectives of the proposed system should be clearly defined and aligned with the organization's strategic goals.

उद्देश्यहरू: प्रस्तावित प्रणालीका उद्देश्यहरू स्पष्ट रूपमा परिभाषित र संगठनको रणनीतिक लक्ष्यहरूसँग पङ्क्तिबद्ध हुनुपर्छ।

Resources: The resources required to develop and implement the proposed system should be identified, including personnel, hardware, software, and budget.

संसाधनहरू: प्रस्तावित प्रणालीको विकास र कार्यान्वयन गर्न आवश्यक स्रोतहरू पहिचान गरिनुपर्छ, जसमा कर्मचारी, हार्डवेयर, सफ्टवेयर र बजेट समावेश हुनुपर्छ।

Types of feasibility: The following types of feasibility should be considered:

Technical feasibility: This involves evaluating whether the proposed system can be developed and implemented using the available technology and resources.

प्राविधिक सम्भाव्यता: यसमा उपलब्ध प्रविधि र स्रोतहरू प्रयोग गरी प्रस्तावित प्रणालीको विकास र कार्यान्वयन गर्न सकिन्छ कि हुँदैन भन्ने मूल्याङ्कन समावेश हुन्छ।

Economic feasibility: This involves evaluating the costs and benefits of the proposed system to determine whether it is financially viable and whether the expected benefits outweigh the costs.

आर्थिक सम्भाव्यता: यसमा प्रस्तावित प्रणालीको लागत र लाभहरू मूल्याङ्कन गरी यो आर्थिक रूपमा व्यवहार्य छ कि छैन र अपेक्षित लाभहरू लागतभन्दा बढी छन् कि छैनन् भनी निर्धारण गर्न समावेश छ।

Operational feasibility: This involves evaluating whether the proposed system is practical and feasible to implement in the organization's existing environment, including the organizational structure, culture, and processes.

परिचालन सम्भाव्यता: यसले संगठनात्मक संरचना, संस्कृति र प्रक्रियाहरू सहित संगठनको अवस्थित वातावरणमा प्रस्तावित प्रणाली व्यावहारिक र कार्यान्वयन गर्न सम्भव छ कि छैन भनेर मूल्याङ्कन समावेश गर्दछ।

Making a judgment on feasibility: Based on the analysis of the objectives, resources, and types of feasibility, a judgment can be made on the feasibility of the proposed system. This judgment should consider the risks, benefits, and costs of the proposed system and should be based on a thorough analysis of the proposed system's potential impact on the organization. If the proposed system is deemed feasible, the system analysis and design process can proceed to the next phase. If the proposed system is deemed not feasible, alternatives should be explored or the project should be terminated.

सम्भाव्यताको आधारमा निर्णय गर्ने: उद्देश्य, स्रोत र सम्भाव्यताका प्रकारहरूको विश्लेषणको आधारमा प्रस्तावित प्रणालीको सम्भाव्यताको आधारमा निर्णय गर्न सकिन्छ। यो निर्णयले प्रस्तावित प्रणालीको जोखिम, लाभ र लागतलाई विचार गर्नुपर्दछ र प्रस्तावित प्रणालीको संगठनमा सम्भावित प्रभावको गहन विश्लेषणमा आधारित हुनुपर्छ। यदि प्रस्तावित प्रणालीलाई सम्भाव्य मानिन्छ भने, प्रणाली विश्लेषण र डिजाइन प्रक्रिया अर्को चरणमा अगाडि बढ्न सक्छ। प्रस्तावित प्रणाली सम्भव नभएको खण्डमा विकल्प खोज्नुपर्छ वा आयोजना खारेज गर्नुपर्छ।

Plan / Control activities

- Time estimation, Gantt charts, CPM and PERT

Planning and controlling activities is an essential part of the system analysis and design process. It involves developing a project plan that outlines the tasks, timelines, resources, and budget required to complete the project successfully. Here are some key techniques used for planning and controlling activities:

योजना र नियन्त्रण गतिविधिहरू प्रणाली विश्लेषण र डिजाइन प्रक्रियाको एक आवश्यक भाग हो। यसले परियोजना योजना विकास गर्न समावेश गर्दछ जसले कार्यहरू, समयरेखाहरू, स्रोतहरू, र परियोजनालाई सफलतापूर्वक पूरा गर्न आवश्यक बजेटको रूपरेखा दिन्छ। यहाँ केही मुख्य प्रविधिहरू योजना र नियन्त्रण गतिविधिहरूको लागि प्रयोग गरिन्छ:

Planning and controlling activities , system analysis and design process को एक आवश्यक भाग हो। यसले project plan विकास गर्न समावेश गर्दछ जसले tasks, timelines, resources, , र Projectलाई successfully (सफलतापूर्वक) पूरा गर्न आवश्यक budget रूपरेखा दिन्छ।

Time estimation: Time estimation involves determining the time required to complete each task in the project plan. This is typically done using historical data, expert opinions, and other methods to estimate how long each task will take.

समय अनुमान: समय अनुमानमा परियोजना योजनामा प्रत्येक कार्य पूरा गर्न आवश्यक समय निर्धारण गर्न समावेश छ। यो सामान्यतया ऐतिहासिक डेटा, विशेषज्ञ राय, र प्रत्येक कार्य कति समय लाग्ने अनुमान गर्न अन्य विधिहरू प्रयोग गरी गरिन्छ।

Time estimation मा project plan मा प्रत्येक कार्य पूरा गर्न time required determine गर्न समावेश छ। यो सामान्यतया historical data, expert opinions, र प्रत्येक कार्य कति समय लाग्ने estimate गर्न अन्य methods प्रयोग गरी गरिन्छ।

Gantt charts: Gantt charts are graphical representations of the project plan that show the timeline for each task and how they are dependent on each other. They are useful for visualizing the project schedule, identifying critical paths, and tracking progress.

Gantt चार्टहरू: Gantt चार्टहरू परियोजना योजनाको ग्राफिकल प्रतिनिधित्व हुन् जसले प्रत्येक कार्यको लागि समयरेखा र कसरी तिनीहरू एकअर्कामा निर्भर छन् भनेर देखाउँछन्। तिनीहरू परियोजना तालिका, महत्वपूर्ण मार्गहरू पहिचान गर्न, र प्रगति ट्र्याकिङको लागि उपयोगी छन्।

Gantt चार्टहरू: Gantt चार्टहरू project planको graphical representations हुन् जसले प्रत्येक taskको लागि timeline र कसरी तिनीहरू एकअर्कामा dependent(निर्भर) छन् भनेर देखाउँछन्। project schedule, identifying critical paths, and tracking progress को लागि उपयोगी छन्।

CPM (Critical Path Method): CPM is a project management technique that identifies the critical path through the project plan. The critical path is the sequence of tasks that must be completed on time to ensure the project is completed on schedule. CPM can be used to determine the earliest possible completion time for a project and to identify tasks that can be delayed without affecting the overall project timeline.

CPM (क्रिटिकल पाथ मेथड): CPM एक परियोजना व्यवस्थापन प्रविधि हो जसले परियोजना योजना माफत महत्वपूर्ण मार्ग पहिचान गर्दछ। महत्वपूर्ण मार्ग भनेको कार्यहरूको क्रम हो जुन परियोजना समयमै पूरा भएको सुनिश्चित गर्नको लागि समयमै पूरा गर्नुपर्छ। CPM लाई परियोजनाको लागि सकेसम्म चाँडो पूरा हुने समय निर्धारण गर्न र परियोजनाको समग्र समयरेखालाई असर नगरी ढिलाइ हुन सक्ने कार्यहरू पहिचान गर्न प्रयोग गर्न सकिन्छ।

CPM एक project management technique हो जसले project plan माफत critical/important path पहिचान गर्दछ। critical path भनेको कार्यहरूको sequence हो जुन project समयमै पूरा भएको ensure

गर्नको लागि समयमै पूरा गर्नुपर्छ। CPM लाई project को लागि सकेसम्म earliest पूरा हुने समय निर्धारण गर्न र projectको समग्र timeline असर नगरी delay हुन सक्ने कार्यहरू पहिचान गर्न प्रयोग गर्न सकिन्छ।

PERT (Program Evaluation and Review Technique): PERT is a project management technique that is like CPM but takes into account the uncertainties and risks associated with each task. PERT uses statistical methods to estimate the time required for each task, and it can be used to identify the probability of completing the project within a given timeframe.

PERT (कार्यक्रम मूल्याङ्कन र समीक्षा प्रविधि): PERT एक परियोजना व्यवस्थापन प्रविधि हो जुन CPM जस्तै हो तर प्रत्येक कार्यसँग सम्बन्धित अनिश्चितता र जोखिमहरूलाई ध्यानमा राख्छ। PERT ले प्रत्येक कार्यको लागि आवश्यक समय अनुमान गर्न सांख्यिकीय विधिहरू प्रयोग गर्दछ, र यसलाई दिइएको समय सीमा भित्र परियोजना पूरा गर्ने सम्भावना पहिचान गर्न प्रयोग गर्न सकिन्छ।

PERT एक project management technique हो जुन CPM जस्तै हो तर प्रत्येक कार्यसँग associated uncertainties and risks ध्यानमा राख्छ। PERT ले प्रत्येक कार्यको लागि आवश्यक समय अनुमान गर्न statistical methods प्रयोग गर्दछ, र यसलाई दिइएको timeframe भित्र project पूरा गर्ने probability पहिचान गर्न प्रयोग गर्न सकिन्छ।

Overall, effective planning and control are crucial to ensuring that the project is completed on time and within budget. Techniques such as time estimation, Gantt charts, CPM, and PERT can help project managers to develop a realistic project plan, identify potential risks, and track progress to ensure that the project is completed successfully.

समग्रमा, परियोजना समयमै र बजेट भित्र सम्पन्न भएको सुनिश्चित गर्न प्रभावकारी योजना र नियन्त्रण महत्वपूर्ण छ। समय अनुमान, Gantt चार्ट, CPM, र PERT जस्ता प्रविधिहरूले परियोजना प्रबन्धकहरूलाई यथार्थपरक परियोजना योजना विकास गर्न, सम्भावित जोखिमहरू पहिचान गर्न र परियोजना सफलतापूर्वक सम्पन्न भएको सुनिश्चित गर्न प्रगति ट्र्याक गर्न मद्दत गर्न सक्छ।

Overall, Project समयमै र within budget सम्पन्न भएको ensure गर्न effective planning and control महत्वपूर्ण छ। Time estimation, Gantt charts, CPM, र PERT जस्ता Techniques ले project managers लाई realistic project plan develop गर्न, potential risks identify गर्न र project successfully complete भएको ensure गर्न process track गर्न मद्दत गर्न सक्छ।

Manage analysis / design activities

- Communication strategies for team management,
- Productivity goal, motivation

- Project failures

Managing analysis and design activities involves effective communication strategies, setting productivity goals, motivating team members, and addressing project failures. Here are some key points to consider:

विश्लेषण र डिजाइन गतिविधिहरू प्रबन्धनमा प्रभावकारी संचार रणनीतिहरू, उत्पादकता लक्ष्यहरू सेट गर्ने, टोलीका सदस्यहरूलाई उत्प्रेरित गर्ने, र परियोजना विफलताहरूलाई सम्बोधन गर्ने समावेश छ। यहाँ विचार गर्न केहि मुख्य बुँदाहरू छन्:

Analysis and design activities Managing मा effective communication strategies, productivity goals setting गर्ने, motivating team members, and addressing project failures समावेश छ। Here are some key points to consider:

Communication strategies for team management: Effective communication is critical to the success of any project. Team members should be clear on their roles and responsibilities, project goals, timelines, and deliverables. Communication tools such as project management software, email, and regular team meetings can help to keep everyone on the same page.

टोली व्यवस्थापनका लागि सञ्चार रणनीतिहरू: प्रभावकारी सञ्चार कुनै पनि परियोजनाको सफलताको लागि महत्वपूर्ण छ। टोलीका सदस्यहरू तिनीहरूको भूमिका र जिम्मेवारीहरू, परियोजना लक्ष्यहरू, समयरेखाहरू, र डेलिभरेबलहरूमा स्पष्ट हुनुपर्छ। सञ्चार उपकरणहरू जस्तै परियोजना व्यवस्थापन सफ्टवेयर, इमेल, र नियमित टोली बैठकहरूले सबैलाई एउटै पृष्ठमा राख्न मद्दत गर्न सक्छ।

Communication strategies for team management (टोली व्यवस्थापनका लागि सञ्चार रणनीतिहरू): Effective communication कुनै पनि project को सफलताको लागि महत्वपूर्ण छ। Team members , तिनीहरूको roles and responsibilities, project goals, timelines, and deliverables मा clear हुनुपर्छ। Communication tools जस्तै project management software, email, and regular team meetings ले सबैलाई एउटै page मा राख्न मद्दत गर्न सक्छ।

Productivity goals and motivation: Setting clear productivity goals and incentives can help to motivate team members and improve their performance. This can include setting deadlines, providing rewards for completing tasks on time, and recognizing and rewarding outstanding performance.

उत्पादकता लक्ष्यहरू र प्रेरणा: स्पष्ट उत्पादकता लक्ष्यहरू र प्रोत्साहनहरू सेट गर्नाले टोलीका सदस्यहरूलाई उत्प्रेरित गर्ने र उनीहरूको प्रदर्शन सुधार गर्न मद्दत गर्न सक्छ। यसले समयसीमाहरू सेट गर्ने, समयमै कार्यहरू पूरा गर्नका लागि पुरस्कारहरू उपलब्ध गराउने, र उत्कृष्ट कार्यसम्पादनलाई चिन्ने र पुरस्कृत गर्ने समावेश गर्न सक्छ।

Productivity goals and motivation (उत्पादकता लक्ष्यहरू र प्रेरणा): clear productivity, goals र incentives सेट गर्नाले team members लाई motivate गर्न र उनीहरूको performance improve गर्न मद्दत गर्न सक्छ। यसले setting deadlines, providing rewards for completing tasks on time, and recognizing and rewarding outstanding performance समावेश गर्न सक्छ।

Project failures: Project failures can occur due to a variety of reasons, including poor planning, miscommunication, lack of resources, and unforeseen obstacles. It's important to identify potential risks and plan for contingencies to avoid project failures. If a project does fail, it's important to conduct a post-mortem analysis to determine what went wrong and how to avoid similar failures in the future.

परियोजना विफलता: परियोजना विफलता विभिन्न कारणहरूले हुन सक्छ, खराब योजना, गलत सञ्चार, स्रोतको कमी, र अप्रत्याशित अवरोधहरू सहित। सम्भावित जोखिमहरू पहिचान गर्न र परियोजना विफलताहरूबाट बच्न आकस्मिकहरूको लागि योजना बनाउनु महत्वपूर्ण छ। यदि कुनै परियोजना असफल भयो भने, के गल्ती भयो र भविष्यमा यस्तै असफलताहरूबाट कसरी बच्ने भन्ने निर्धारण गर्न पोस्टमार्टम विश्लेषण गर्नु महत्वपूर्ण हुन्छ।

Project failures विभिन्न कारणहरूले हुन सक्छ, including poor planning, miscommunication, lack of resources, and unforeseen obstacles । potential risks पहिचान र project failures बाट बच्न contingencies(आकस्मिकहरू) को लागि plan बनाउनु महत्वपूर्ण छ। यदि कुनै project fail भयो भने, के गल्ती भयो र future मा यस्तै failuresबाट कसरी बच्ने भन्ने निर्धारण गर्न post-mortem analysis गर्नु महत्वपूर्ण हुन्छ।

Overall, managing analysis and design activities requires effective communication, clear productivity goals, motivation, and a plan for addressing potential risks and failures. By implementing these strategies, project managers can ensure that their teams are productive, motivated, and successful in delivering high-quality systems.

समग्रमा, विश्लेषण र डिजाइन गतिविधिहरू प्रबन्ध गर्न प्रभावकारी सञ्चार, स्पष्ट उत्पादकता लक्ष्यहरू, प्रेरणा, र सम्भावित जोखिमहरू र विफलताहरूलाई सम्बोधन गर्न योजना आवश्यक पर्दछ। यी रणनीतिहरू लागू गरेर, परियोजना प्रबन्धकहरूले उनीहरूको टोलीहरू उत्पादक, उत्प्रेरित, र उच्च-गुणस्तर प्रणालीहरू प्रदान गर्न सफल छन् भनी सुनिश्चित गर्न सक्छन्।

Overall, Analysis and design activities Managing मा effective communication, clear productivity goals, motivation, र potential risks and failures लाई address गर्न plan आवश्यक पर्दछ। यी strategies लागू गरेर, project managers ले उनीहरूको teams productive, motivated, र high-quality systems प्रदान गर्न सफल छन् भनी सुनिश्चित गर्न सक्छन्।

Module 3: Collect data

Design sample

- Need of sampling
- Types of sampling; Convenience, Purposive, Random
- Sample size

Designing a sample is an important aspect of data collection in system analysis and design. Here are some key points to consider:

नमूना डिजाइन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक महत्वपूर्ण पक्ष हो। यहाँ विचार गर्न केहि मुख्य बुँदाहरू छन्:

Sample Designing, System Analysis र Design मा data collection को एक महत्वपूर्ण aspect हो। Here are some key points to consider:

Need of sampling: Sampling is a method of selecting a subset of the population to be studied. This is done when it is not feasible or practical to study the entire population. Sampling can help to save time and resources while still providing accurate information.

नमूनाको आवश्यकता: नमूना अध्ययन गरिनु पर्ने जनसंख्याको उपसमूह चयन गर्ने विधि हो। यो सम्पूर्ण जनसंख्या अध्ययन गर्न सम्भव वा व्यावहारिक नभएको बेला गरिन्छ। नमूनाले अझै पनि सही जानकारी प्रदान गर्दा समय र स्रोतहरू बचत गर्न मद्दत गर्न सक्छ।

Sampling अध्ययन गरिनु पर्ने population(जनसंख्या) को subset(उपसमूह) select गर्ने method हो। यो entire population study गर्न feasible वा practical नभएको बेला गरिन्छ। Sampling ले अझै पनि accurate information प्रदान गर्दा time र resource बचत(save) गर्न मद्दत गर्न सक्छ।

Types of sampling: There are several types of sampling methods including convenience sampling, purposive sampling, and random sampling. **Convenience sampling** involves selecting individuals who are easily accessible. **Purposive sampling** involves selecting individuals based on specific criteria. **Random sampling** involves selecting individuals at random from the population.

नमूनाका प्रकारहरू: त्यहाँ धेरै प्रकारका नमूना विधिहरू छन् जसमा सुविधा नमूना, उद्देश्यपूर्ण नमूना, र अनियमित नमूनाहरू समावेश छन्। सुविधा नमूनामा सजिलै पहुँचयोग्य व्यक्तिहरू छनौट गर्ने समावेश छ। उद्देश्यपूर्ण नमूनाले विशिष्ट मापदण्डको आधारमा व्यक्तिहरूको चयन समावेश गर्दछ। अनियमित नमूनाले जनसंख्याबाट अनियमित रूपमा व्यक्तिहरू चयन गर्ने समावेश गर्दछ।

त्यहाँ धेरै types का sampling methods छन् convenience sampling, purposive sampling, and random sampling समावेश छन्। convenience sampling मा easily(सजिलै) accessible(पहुँचयोग्य) व्यक्तिहरू छनौट गर्ने समावेश छ। purposive sampling ले specific criteria को आधारमा व्यक्तिहरूको

चयन समावेश गर्दछ। Random sampling ले population बाट random रूपमा व्यक्तिहरू चयन गर्ने समावेश गर्दछ।

Sample size: The sample size is an important consideration when designing a sample. A larger sample size can provide more accurate results, but it can also be more time-consuming and expensive to collect. The sample size should be large enough to provide a representative sample of the population, but small enough to be practical.

नमूना आकार: नमूना डिजाइन गर्दा नमूना आकार महत्वपूर्ण विचार हो। ठूलो नमूना आकारले थप सटीक परिणामहरू प्रदान गर्न सक्छ, तर यो सङ्कलन गर्न धेरै समय खपत र महँगो पनि हुन सक्छ। नमूना आकार जनसंख्याको प्रतिनिधि नमूना प्रदान गर्न पर्याप्त ठूलो हुनुपर्छ, तर व्यावहारिक हुन पर्याप्त सानो।

Sample Design गर्दा Sample size महत्वपूर्ण विचार हो। Large sample sizeले more accurate results प्रदान गर्न सक्छ, तर यो collect गर्न धेरै समय खपत र महँगो पनि हुन सक्छ। Sample size population को representative sample प्रदान गर्न enough large हुनुपर्छ, तर practical हुन सानो पर्याप्त xa.

Overall, designing a sample involves selecting a sampling method that is appropriate for the research question and population of interest, determining the sample size, and selecting individuals to participate in the study. By designing a representative sample, system analysts and designers can obtain accurate and reliable data to inform their work.

समग्रमा, नमूना डिजाइन गर्दा अनुसन्धान प्रश्न र रुचिको जनसंख्याको लागि उपयुक्त नमूना विधि चयन गर्ने, नमूना आकार निर्धारण गर्ने, र अध्ययनमा भाग लिन व्यक्तिहरू छनौट गर्ने समावेश छ। एक प्रतिनिधि नमूना डिजाइन गरेर, प्रणाली विश्लेषक र डिजाइनरहरूले आफ्नो काम सूचित गर्न सही र भरपर्दो डाटा प्राप्त गर्न सक्छन्।

Overall, Sample Designing गर्दा research question र interestको population को लागि appropriate sample method select गर्ने, Sample size determine गर्ने, र study मा participate लिन individuals select गर्ने समावेश छ। representative sample design गरेर, system analysts and designers ले आफ्नो काम सूचित गर्न accurate र reliable डाटा प्राप्त गर्न सक्छन्।

Conduct interview

- Planning interview, Objectives of interview

- Types of question; Close & Open ended, Probe
- Question pitfalls

Conducting interviews is a common method of data collection in system analysis and design. Here are some key points to consider:

अन्तर्वार्ता सञ्चालन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक सामान्य विधि हो। यहाँ विचार गर्न केहि मुख्य बुँदाहरू छन्:

Planning interview: Planning an interview involves determining the objectives of the interview, identifying the individuals to be interviewed, and deciding on the format and structure of the interview.

अन्तर्वार्ताको योजना बनाउने: अन्तर्वार्ताको योजना बनाउनुमा अन्तर्वार्ताको उद्देश्यहरू निर्धारण गर्ने, अन्तर्वार्ता लिने व्यक्तिहरूको पहिचान गर्ने र अन्तर्वार्ताको ढाँचा र संरचनाबारे निर्णय गर्ने समावेश हुन्छ।

Objectives of interview: The objectives of the interview should be clear and well-defined. These may include gathering information about the system being analyzed, understanding user needs and requirements, or evaluating the effectiveness of a system.

अन्तर्वार्ताका उद्देश्यहरू: अन्तर्वार्ताका उद्देश्यहरू स्पष्ट र राम्रोसँग परिभाषित हुनुपर्छ। यसमा विश्लेषण भइरहेको प्रणालीको बारेमा जानकारी सङ्कलन, प्रयोगकर्ताको आवश्यकता र आवश्यकताहरू बुझ्न, वा प्रणालीको प्रभावकारिता मूल्याङ्कन समावेश हुन सक्छ।

Types of questions: There are two main types of questions that can be used in an interview: closed-ended and open-ended. Closed-ended questions require a specific response and are useful for gathering quantitative data. Open-ended questions allow the interviewee to provide a more detailed response and are useful for gathering qualitative data. Probing questions can also be used to encourage the interviewee to provide more detailed responses.

प्रश्नहरूको प्रकार: त्यहाँ दुई मुख्य प्रकारका प्रश्नहरू छन् जुन अन्तर्वार्तामा प्रयोग गर्न सकिन्छ: बन्द-अन्त र खुला-अन्त। बन्द-समाप्त प्रश्नहरूलाई एक विशेष प्रतिक्रिया चाहिन्छ र मात्रात्मक डेटा सङ्कलन गर्न उपयोगी हुन्छ। खुला-समाप्त प्रश्नहरूले अन्तर्वार्ता लिनेहरूलाई थप विस्तृत प्रतिक्रिया प्रदान गर्न अनुमति दिन्छ र गुणात्मक डेटा सङ्कलन गर्न उपयोगी हुन्छ। थप विस्तृत प्रतिक्रियाहरू प्रदान गर्न अन्तर्वार्ता लिनेलाई प्रोत्साहित गर्नका लागि प्रश्नहरू जाँच्न पनि प्रयोग गर्न सकिन्छ।

Question pitfalls: It is important to be aware of potential pitfalls in interview questions, such as leading questions that bias the response, double-barrelled questions that ask about two things at once, or complex questions that may confuse the interviewee.

प्रश्नका समस्याहरू: अन्तर्वार्ता प्रश्नहरूमा सम्भावित कमजोरीहरू बारे सचेत हुनु महत्वपूर्ण छ, जस्तै प्रतिक्रियालाई पूर्वाग्रह गर्ने प्रमुख प्रश्नहरू, एकै पटक दुईवटा कुराहरू सोध्ने डबल ब्यारेल प्रश्नहरू, वा अन्तर्वार्ता लिनेलाई भ्रमित पार्न सक्ने जटिल प्रश्नहरू।

Overall, conducting interviews can be a valuable method of data collection in system analysis and design. By carefully planning the interview, using appropriate types of questions, and avoiding question pitfalls, system analysts and designers can gather accurate and reliable data to inform their work.

समग्रमा, अन्तर्वार्ता सञ्चालन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक बहुमूल्य विधि हुन सक्छ। सावधानीपूर्वक अन्तर्वार्ताको योजना बनाएर, उपयुक्त प्रकारका प्रश्नहरू प्रयोग गरेर, र प्रश्नका समस्याहरूलाई बेवास्ता गरेर, प्रणाली विश्लेषकहरू र डिजाइनरहरूले आफ्नो कामको जानकारी दिन सही र भरपर्दो डाटा सङ्कलन गर्न सक्छन्।

Administer questionnaires

- Planning questionnaire administration
- Scales in questionnaire
- Validity, Reliability
- Clustering of questions of similar content

Administering questionnaires is another common method of data collection in system analysis and design. Here are some key points to consider:

प्रणाली विश्लेषण र डिजाइनमा डाटा सङ्कलन गर्ने अर्को सामान्य विधि प्रश्नावलीहरू व्यवस्थापन गर्ने हो। यहाँ विचार गर्न केहि मुख्य बुँदाहरू छन्:

Planning questionnaire administration: Planning questionnaire administration involves identifying the target population, determining the purpose of the questionnaire, and designing the questionnaire itself. It is important to ensure that the questions are clear and unambiguous and that they address the research objectives.

योजना प्रश्नावली व्यवस्थापन: योजना प्रश्नावली व्यवस्थापनमा लक्षित जनसंख्या पहिचान गर्ने, प्रश्नावलीको उद्देश्य निर्धारण गर्ने र प्रश्नावली आफैं डिजाइन गर्ने समावेश छ। यो सुनिश्चित गर्न महत्वपूर्ण छ कि प्रश्नहरू स्पष्ट र अस्पष्ट छन् र तिनीहरूले अनुसन्धान उद्देश्यहरूलाई सम्बोधन गर्छन्।

Scales in questionnaire: Scales can be used in questionnaires to measure variables such as attitudes, beliefs, or opinions. Common types of scales include Likert scales, semantic differential scales, and

visual analogue scales. It is important to select the appropriate type of scale for the research objectives and to ensure that the scale is used consistently throughout the questionnaire.

प्रश्नावलीमा स्केलहरू: तराजूहरू प्रश्नावलीहरूमा व्यवहार, विश्वास, वा विचारहरू जस्ता चरहरू मापन गर्न प्रयोग गर्न सकिन्छ। साधारण प्रकारका तराजूहरूमा लिकर्ट स्केल, सिमेन्टिक डिफरेंशियल स्केल र भिजुअल एनालोग स्केलहरू समावेश छन्। अनुसन्धान उद्देश्यहरूको लागि उपयुक्त प्रकारको मापन चयन गर्न र प्रश्नावली भर मापन निरन्तर रूपमा प्रयोग भएको सुनिश्चित गर्न महत्वपूर्ण छ।

Validity and reliability: Validity refers to the extent to which a questionnaire measures what it is intended to measure. Reliability refers to the consistency of the questionnaire over time and across different groups of respondents. To ensure validity and reliability, it is important to pilot test the questionnaire and to use established measures whenever possible.

वैधता र विश्वसनीयता: वैधताले प्रश्नावलीले मापन गर्ने उद्देश्यले के मापन गर्छ भनेर बुझाउँछ। विश्वसनीयताले समय र उत्तरदाताहरूको विभिन्न समूहहरूमा प्रश्नावलीको एकरूपतालाई जनाउँछ। वैधता र विश्वसनीयता सुनिश्चित गर्न, प्रश्नावलीको परीक्षण परीक्षण गर्न र सम्भव भएसम्म स्थापित उपायहरू प्रयोग गर्न महत्वपूर्ण छ।

Clustering of questions of similar content: Clustering questions of similar content can help to ensure that the questionnaire is organized and easy to complete. Clustering can also help to identify patterns in the data and to analyze the results more effectively.

समान सामग्रीका प्रश्नहरूको क्लस्टरिङ: समान सामग्रीका प्रश्नहरूको क्लस्टरिङले प्रश्नावली व्यवस्थित र पूरा गर्न सजिलो छ भनी सुनिश्चित गर्न मद्दत गर्न सक्छ। क्लस्टरिङले डाटामा ढाँचाहरू पहिचान गर्न र परिणामहरूलाई अझ प्रभावकारी रूपमा विश्लेषण गर्न मद्दत गर्न सक्छ।

Overall, administering questionnaires can be an effective method of data collection in system analysis and design. By carefully planning the questionnaire administration, using appropriate scales, ensuring validity and reliability, and clustering questions of similar content, system analysts and designers can gather accurate and reliable data to inform their work.

समग्रमा, प्रश्नावली व्यवस्थापन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक प्रभावकारी तरीका हुन सक्छ। प्रश्नावली व्यवस्थापनको सावधानीपूर्वक योजना बनाएर, उपयुक्त स्केलहरू प्रयोग गरेर, वैधता र विश्वसनीयता सुनिश्चित गर्दै, र समान सामग्रीका प्रश्नहरूको क्लस्टरिङ गरेर, प्रणाली विश्लेषकहरू र डिजाइनरहरूले आफ्नो कामको जानकारी दिन सही र भरपर्दो डाटा सङ्कलन गर्न सक्छन्।

Observe decision making activity

- Time event sampling, body language of decision maker

निर्णय गर्ने गतिविधि अवलोकन गर्नुहोस्

- समय घटना नमूना, निर्णय निर्माता को शरीर भाषा

Observing decision-making activities is another method of data collection in system analysis and design. Here are some key points to consider:

निर्णय लिने गतिविधिहरू अवलोकन गर्नु प्रणाली विश्लेषण र डिजाइनमा डाटा सङ्कलन गर्ने अर्को विधि हो। यहाँ विचार गर्ने केहि मुख्य बुँदाहरू छन्:

Time event sampling: Time event sampling involves observing the decision-making process and recording what is happening at predetermined intervals. For example, an observer might record what the decision maker is doing every 5 minutes. Time event sampling can help to identify patterns in the decision-making process and to identify areas where improvements can be made.

समय घटना नमूना: समय घटना नमूना निर्णय प्रक्रिया अवलोकन र पूर्वनिर्धारित अन्तरालहरूमा के भइरहेको छ रेकर्ड गर्न समावेश छ। उदाहरणका लागि, एक पर्यवेक्षकले प्रत्येक 5 मिनेटमा निर्णय निर्माताले के गरिरहेको छ भनेर रेकर्ड गर्न सक्छ। समय घटना नमूनाले निर्णय प्रक्रियामा ढाँचाहरू पहिचान गर्न र सुधारहरू गर्न सकिने क्षेत्रहरू पहिचान गर्न मद्दत गर्न सक्छ।

Body language of decision maker: Observing the body language of the decision maker can provide important insights into their thought processes and emotional state. For example, a decision maker who is leaning forward and making eye contact may be more engaged in the decision-making process than one who is slouching and avoiding eye contact. Observing body language can help to identify the decision maker's level of interest, confidence, and engagement in the decision-making process.

निर्णय निर्माताको शारीरिक भाषा: निर्णय निर्माताको शारीरिक भाषा अवलोकन गर्नाले उनीहरूको विचार प्रक्रिया र भावनात्मक अवस्थाको महत्वपूर्ण अन्तर्दृष्टि प्रदान गर्न सक्छ। उदाहरण को लागी, एक निर्णय निर्माता जो अगाडि झुकिरहेको छ र आँखा सम्पर्क गर्दैछ त्यो निर्णय लिने प्रक्रियामा अधिक संलग्न हुन सक्छ जसले आँखाको सम्पर्कलाई बेवास्ता गरिरहेको छ। शारीरिक भाषा अवलोकन गर्नाले निर्णय लिने प्रक्रियामा निर्णयकर्ताको चासो, आत्मविश्वास र संलग्नताको स्तर पहिचान गर्न मद्दत गर्न सक्छ।

Overall, observing decision-making activities can be an effective method of data collection in system analysis and design. By using time event sampling and observing the body language of the decision maker, system analysts and designers can gather valuable insights into the decision-making process and use this information to inform their work.

समग्रमा, निर्णय लिने गतिविधिहरू अवलोकन गर्नु प्रणाली विश्लेषण र डिजाइनमा डाटा सङ्कलनको प्रभावकारी विधि हुन सक्छ। समय घटना नमूना प्रयोग गरेर र निर्णय निर्माताको शरीर भाषा अवलोकन गरेर, प्रणाली विश्लेषकहरू र डिजाइनरहरूले निर्णय प्रक्रियामा बहुमूल्य अन्तर्दृष्टिहरू भेला गर्न सक्छन् र उनीहरूको कामलाई सूचित गर्न यो जानकारी प्रयोग गर्न सक्छन्।

Observe office environment

- Filmic & Organisational elements, STROBE (STructured Observation of the Environment)

Observing the office environment can be an important method of data collection in system analysis and design. Here are some key points to consider:

Filmic and organizational elements: Observing the office environment involves paying attention to both the physical and organizational elements of the workplace. Physical elements may include the layout of the space, the furniture and equipment used, and the lighting and sound quality. Organizational elements may include the communication and collaboration processes used, the level of employee engagement and satisfaction, and the overall culture and values of the organization. By observing these elements, system analysts and designers can gain a better understanding of how the office environment supports or hinders the work being done.

STROBE: STROBE (Structured Observation of the Environment) is a method of observing the office environment that involves using a structured framework to collect data. This framework includes a set of predefined categories and subcategories that cover different aspects of the office environment, such as the physical layout, the communication processes, and the level of collaboration among employees. Using the STROBE framework can help to ensure that data is collected consistently and systematically, making it easier to analyze and interpret.

Overall, observing the office environment can be a valuable method of data collection in system analysis and design. By paying attention to both the physical and organizational elements of the workplace and using a structured framework such as STROBE, system analysts and designers can gather important insights into how the office environment supports or hinders the work being done, and use this information to inform their work.

कार्यालयको वातावरण नियाल्नुहोस्

- चलचित्र र संगठनात्मक तत्व, स्ट्रोब (वातावरणको संरचनात्मक अवलोकन)

कार्यालय वातावरण अवलोकन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक महत्वपूर्ण विधि हुन सक्छ। यहाँ विचार गर्न केहि मुख्य बुँदाहरू छन्:

चलचित्र र संगठनात्मक तत्वहरू: कार्यालयको वातावरण अवलोकन गर्दा कार्यस्थलको भौतिक र संगठनात्मक तत्वहरूमा ध्यान दिनु समावेश छ। भौतिक तत्वहरूले ठाउँको लेआउट, प्रयोग गरिएका फर्निचर र उपकरणहरू, र प्रकाश र ध्वनि गुणस्तर समावेश गर्न सक्छन्। संगठनात्मक तत्वहरूले प्रयोग गरिएका सञ्चार र सहकार्य प्रक्रियाहरू, कर्मचारी संलग्नता र सन्तुष्टिको स्तर, र संगठनको समग्र संस्कृति र मूल्यहरू समावेश गर्न

सकछन्। यी तत्वहरू अवलोकन गरेर, प्रणाली विश्लेषकहरू र डिजाइनरहरूले कार्यालयको वातावरणले काम गरिरहेको कामलाई कसरी समर्थन वा बाधा पुऱ्याउँछ भन्ने बारे राम्रोसँग बुझ्न सकछन्।

स्ट्रोब: स्ट्रोब (वातावरणको संरचित अवलोकन) कार्यालयको वातावरण अवलोकन गर्ने एक विधि हो जसमा तथ्याङ्क सङ्कलन गर्ने संरचित फ्रेमवर्क प्रयोग गरिन्छ। यस ढाँचामा पूर्वनिर्धारित कोटीहरू र उपश्रेणीहरूको सेट समावेश छ जसले कार्यालय वातावरणका विभिन्न पक्षहरू समावेश गर्दछ, जस्तै भौतिक रूपरेखा, सञ्चार प्रक्रियाहरू, र कर्मचारीहरू बीचको सहकार्यको स्तर। STROBE ढाँचाको प्रयोगले यो सुनिश्चित गर्न मद्दत गर्न सकछ कि डेटा लगातार र व्यवस्थित रूपमा सङ्कलन गरिएको छ, यसलाई विश्लेषण र व्याख्या गर्न सजिलो बनाउँछ।

समग्रमा, कार्यालय वातावरण अवलोकन प्रणाली विश्लेषण र डिजाइन मा डाटा संग्रह को एक बहुमूल्य विधि हुन सकछ। कार्यस्थलका दुवै भौतिक र संगठनात्मक तत्वहरूमा ध्यान दिएर र STROBE जस्ता संरचित ढाँचा प्रयोग गरेर, प्रणाली विश्लेषकहरू र डिजाइनरहरूले कार्यालयको वातावरणले कसरी भइरहेको कामलाई समर्थन गर्छ वा बाधा पुऱ्याउँछ भन्ने बारे महत्वपूर्ण अन्तरदृष्टिहरू सङ्कलन गर्न सकछन्, र तिनीहरूको कामलाई सूचित गर्न यो जानकारी प्रयोग गर्न सकछन्।

Develop prototype / model

- Types of Prototypes; Patched up, Non-operational, First -of-a-series, selected featured
- Guidelines for developing a prototypes
- Modifications on user's interface
- Advantage and disadvantage of prototypes

A prototype is a preliminary model or sample of a system or product that is being developed. Its main purpose is to allow users and developers to explore and test the system's functionality, usability, and design before the final version is completed. The following are some key points related to developing a prototype/model:

Types of prototypes: There are several types of prototypes, including patched-up, non-operational, first-of-a-series, and selected featured. **A patched-up** prototype is a quick and simple model that can be easily put together. **A non-operational** prototype is one that has the look and feel of the final product but does not function. A **first-of-a-series** prototype is a more refined version of a non-operational prototype, and it is used to test and refine the system's functionality. A **selected featured** prototype is a combination of the above prototypes and is used to test specific features of the system.

Guidelines for developing a prototype: The following are some guidelines for developing a prototype: identify the purpose and scope of the prototype, select the appropriate type of prototype, involve users in the design and testing process, use existing software and hardware where possible, and document the prototype development process.

Modifications on user's interface: One of the key benefits of a prototype is that it allows developers to make modifications to the user interface based on feedback from users. This can help to ensure that the final product is user-friendly and meets the needs of the intended audience.

Advantage and disadvantage of prototypes: The main advantage of a prototype is that it allows developers to test and refine the system's functionality and design before the final version is

completed. This can help to identify and address potential issues early in the development process, which can save time and resources. However, prototypes can be time-consuming and expensive to develop, and there is a risk that users may become attached to certain features of the prototype that cannot be included in the final product.

प्रोटोटाइप / मोडेल विकास गर्नुहोस्

- प्रोटोटाइपका प्रकारहरू; प्याच अप, गैर-अपरेसनल, पहिलो-अफ-ए-श्रृंखला, चयन गरिएको विशेष
- प्रोटोटाइपहरू विकास गर्नका लागि दिशानिर्देशहरू
- प्रयोगकर्ताको इन्टरफेसमा परिमार्जनहरू
- प्रोटोटाइपको फाइदा र बेफाइदा

एक प्रोटोटाइप एक प्रारम्भिक मोडेल वा प्रणाली वा उत्पादन को नमूना हो जुन विकसित भइरहेको छ। यसको मुख्य उद्देश्य प्रयोगकर्ताहरू र विकासकर्ताहरूलाई प्रणालीको कार्यक्षमता, उपयोगिता, र डिजाइनको अन्तिम संस्करण पूरा हुनु अघि अन्वेषण र परीक्षण गर्न अनुमति दिनु हो। प्रोटोटाइप/मोडेलको विकाससँग सम्बन्धित केही मुख्य बुँदाहरू निम्न छन्:

प्रोटोटाइपका प्रकारहरू: प्याच-अप, गैर-अपरेसनल, फर्स्ट-अफ-ए-श्रृंखला, र चयन गरिएका विशेषताहरू सहित धेरै प्रकारका प्रोटोटाइपहरू छन्। एक प्याच-अप प्रोटोटाइप एक द्रुत र सरल मोडेल हो जुन सजिलै संग राख्न सकिन्छ। एक गैर-अपरेसनल प्रोटोटाइप एक हो जसमा अन्तिम उत्पादनको रूप र अनुभव छ तर कार्य गर्दैन। पहिलो-को-ए-श्रृंखला प्रोटोटाइप गैर-अपरेसनल प्रोटोटाइपको अधिक परिष्कृत संस्करण हो, र यो प्रणालीको कार्यक्षमता परीक्षण र परिष्कृत गर्न प्रयोग गरिन्छ। चयन गरिएको विशेष प्रोटोटाइप माथिको प्रोटोटाइपहरूको संयोजन हो र प्रणालीको विशिष्ट सुविधाहरू परीक्षण गर्न प्रयोग गरिन्छ।

प्रोटोटाइप विकास गर्ने दिशानिर्देशहरू: प्रोटोटाइप विकास गर्नका लागि निम्न केही दिशानिर्देशहरू छन्: प्रोटोटाइपको उद्देश्य र दायरा पहिचान गर्नुहोस्, उपयुक्त प्रकारको प्रोटोटाइप चयन गर्नुहोस्, प्रयोगकर्ताहरूलाई डिजाइन र परीक्षण प्रक्रियामा संलग्न गर्नुहोस्, सम्भव भएसम्म अवस्थित सफ्टवेयर र हार्डवेयर प्रयोग गर्नुहोस्, र प्रोटोटाइप विकास प्रक्रिया कागजात गर्नुहोस्।

प्रयोगकर्ताको इन्टरफेसमा परिमार्जनहरू: प्रोटोटाइपको मुख्य फाइदाहरू मध्ये एक यो हो कि यसले विकासकर्ताहरूलाई प्रयोगकर्ताहरूको प्रतिक्रियाको आधारमा प्रयोगकर्ता इन्टरफेसमा परिमार्जन गर्न अनुमति दिन्छ। यसले अन्तिम उत्पादन प्रयोगकर्ता-अनुकूल छ र लक्षित दर्शकहरूको आवश्यकताहरू पूरा गर्दछ भनेर सुनिश्चित गर्न मद्दत गर्न सक्छ।

प्रोटोटाइपको फाइदा र हानि: प्रोटोटाइपको मुख्य फाइदा यो हो कि यसले विकासकर्ताहरूलाई अन्तिम संस्करण पूरा हुनु अघि प्रणालीको कार्यक्षमता र डिजाइनलाई परीक्षण र परिष्कृत गर्न अनुमति दिन्छ। यसले विकास प्रक्रियाको प्रारम्भमा सम्भावित समस्याहरू पहिचान गर्न र सम्बोधन गर्न मद्दत गर्न सक्छ, जसले समय र

स्रोतहरू बचत गर्न सक्छ। यद्यपि, प्रोटोटाइपहरू विकास गर्न समय-खपत र महँगो हुन सक्छ, र प्रयोगकर्ताहरू अन्तिम उत्पादनमा समावेश गर्न नसकिने प्रोटोटाइपका निश्चित सुविधाहरूमा संलग्न हुन सक्ने जोखिम छ।

Module 4: Analyze /Process data

Develop data flow diagram

- Basic symbols of data flow, conventions
- Context diagram
- Approaches in data flow diagram
- Labelling

A **data flow diagram (DFD)** is a graphical representation of the flow of data through a system. The development of a DFD is an important step in system analysis and design. The following are some of the key topics that need to be covered while developing a DFD:

Basic symbols of data flow: A DFD consists of four basic symbols: circles, squares, arrows, and lines. Circles represent entities or processes that are involved in the system, squares represent data stores where the data is stored, arrows represent data flows, and lines represent the boundaries of the system.

Conventions: There are certain conventions that need to be followed while developing a DFD, such as each process should have at least one input and one output, a data store should have at least one input or output, and data should flow from higher-level processes to lower-level processes.

Context diagram: A context diagram is a high-level view of the system that shows the system as a single process with inputs and outputs. It is used to identify the system boundaries and the external entities that interact with the system.

Approaches in data flow diagram: There are two approaches to developing a DFD: top-down and bottom-up. In the top-down approach, the system is divided into subsystems and each subsystem is further analyzed until the required level of detail is reached. In the bottom-up approach, the subsystems are analyzed first and then combined to form the overall system.

Labelling: Each data flow, process, and data store should be labelled appropriately to clearly identify its purpose and function within the system.

Overall, a well-designed DFD can help in understanding the flow of data through the system, identifying bottlenecks, and improving the overall efficiency of the system.

मोड्युल ४: विश्लेषण/प्रक्रिया डाटा

डाटा प्रवाह रेखाचित्र विकास गर्नुहोस्

- डाटा प्रवाहको आधारभूत प्रतीकहरू, सम्मेलनहरू
- सन्दर्भ रेखाचित्र
- डाटा प्रवाह रेखाचित्रमा दृष्टिकोण

• लेबलिङ

डाटा प्रवाह रेखाचित्र (DFD) प्रणाली मार्फत डाटा को प्रवाह को एक ग्राफिकल प्रतिनिधित्व हो। DFD को विकास प्रणाली विश्लेषण र डिजाइन मा एक महत्वपूर्ण कदम हो। DFD विकास गर्दा कभर गर्नुपर्ने केही मुख्य विषयहरू निम्न छन्:

डाटा प्रवाहको आधारभूत प्रतीकहरू: DFD मा चारवटा आधारभूत प्रतीकहरू हुन्छन्: सर्कल, वर्ग, तीर र रेखाहरू। सर्कलहरूले प्रणालीमा संलग्न निकाय वा प्रक्रियाहरूलाई प्रतिनिधित्व गर्दछ, वर्गहरूले डेटा भण्डारण गर्ने डेटा भण्डारहरू प्रतिनिधित्व गर्दछ, तीरहरूले डेटा प्रवाहलाई प्रतिनिधित्व गर्दछ, र रेखाहरूले प्रणालीको सीमाहरू प्रतिनिधित्व गर्दछ।

कन्भेन्सनहरू: DFD विकास गर्दा पछ्याउन आवश्यक केही कन्भेन्सनहरू छन्, जस्तै प्रत्येक प्रक्रियामा कम्तिमा एउटा इनपुट र एउटा आउटपुट हुनुपर्छ, डाटा स्टोरमा कम्तिमा एउटा इनपुट वा आउटपुट हुनुपर्छ, र डाटा उच्च-स्तर प्रक्रियाहरूबाट तल्लो-स्तर प्रक्रियाहरूमा प्रवाह गर्नुपर्छ।

सन्दर्भ रेखाचित्र: एक सन्दर्भ रेखाचित्र प्रणालीको उच्च-स्तरको दृश्य हो जसले प्रणालीलाई इनपुट र आउटपुटहरूको साथ एकल प्रक्रियाको रूपमा देखाउँछ। यो प्रणाली सीमाहरू र प्रणालीसँग अन्तरक्रिया गर्ने बाह्य संस्थाहरू पहिचान गर्न प्रयोग गरिन्छ।

डेटा प्रवाह रेखाचित्रमा दृष्टिकोणहरू: DFD विकास गर्न दुईवटा दृष्टिकोणहरू छन्: माथि-डाउन र तल-माथि। माथि-डाउन दृष्टिकोणमा, प्रणालीलाई उपप्रणालीहरूमा विभाजन गरिएको छ र आवश्यक स्तरको विवरण नपुगेसम्म प्रत्येक उपप्रणालीलाई थप विश्लेषण गरिन्छ। तल्लो-माथिको दृष्टिकोणमा, उपप्रणालीहरूलाई पहिले विश्लेषण गरिन्छ र त्यसपछि समग्र प्रणाली बनाउनको लागि जोडिन्छ।

लेबलिङ: प्रत्येक डाटा प्रवाह, प्रक्रिया, र डाटा भण्डारलाई प्रणाली भित्र यसको उद्देश्य र कार्य स्पष्ट रूपमा पहिचान गर्न उपयुक्त रूपमा लेबल गरिनुपर्छ।

समग्रमा, राम्रोसँग डिजाइन गरिएको DFD ले प्रणाली मार्फत डाटाको प्रवाह बुझ्न, अवरोधहरू पहिचान गर्न, र प्रणालीको समग्र दक्षता सुधार गर्न मद्दत गर्न सक्छ।

Develop data dictionary

- Data in data dictionary,
- Cataloguing,
- Data processes, data flow, data store, data structure, data elements
- Steps in compiling data dictionaries

A data dictionary is a central repository of information about data that is used within an organization. It provides a description of the data elements, data structures, data flows, and data stores in a system. The following are the steps involved in compiling a data dictionary:

Data in data dictionary: The data dictionary should include information about all data elements used in the system. This information should include data element names, descriptions, data types, lengths, formats, and other properties.

Cataloguing: Cataloguing is the process of organizing and grouping data elements in a logical way to make it easier to manage and maintain the data dictionary. This involves creating categories and subcategories of data elements.

Data processes: Data processes are the steps involved in creating, updating, retrieving, or deleting data in the system. The data dictionary should include information about all data processes and their associated data elements.

Data flow: Data flow describes the movement of data through the system. The data dictionary should include information about all data flows and their associated data elements.

Data store: Data store is where data is stored in the system. The data dictionary should include information about all data stores and their associated data elements.

Data structure: Data structure refers to the way data is organized within a data store. The data dictionary should include information about all data structures used in the system.

Data elements: Data elements are the basic units of information used in the system. The data dictionary should include detailed information about all data elements used in the system.

Steps in compiling data dictionaries: The process of compiling a data dictionary involves the following steps:

- Identify all data elements used in the system
- Catalogue data elements into logical categories and subcategories
- Define data elements and their properties, such as data type, length, format, and other attributes
- Document data flows, data stores, and data structures
- Validate and verify the accuracy of the data dictionary
- Update the data dictionary as changes are made to the system.

By following these steps, a comprehensive data dictionary can be created that provides a detailed understanding of the data used in a system, which can be useful for system analysis, design, and maintenance.

डाटा शब्दकोश विकास गर्नुहोस्

- डाटा शब्दकोशमा डाटा,
- सूचीकरण,
- डाटा प्रक्रियाहरू, डाटा प्रवाह, डाटा भण्डार, डाटा संरचना, डाटा तत्वहरू
- डेटा शब्दकोशहरू संकलन गर्ने चरणहरू

डाटा शब्दकोश भनेको संगठन भित्र प्रयोग हुने डाटाको बारेमा जानकारीको केन्द्रीय भण्डार हो। यसले डाटा तत्वहरू, डाटा संरचनाहरू, डाटा प्रवाहहरू, र प्रणालीमा डाटा भण्डारहरूको विवरण प्रदान गर्दछ। डाटा शब्दकोश कम्पाइल गर्न निम्न चरणहरू समावेश छन्:

डाटा शब्दकोशमा डाटा: डाटा शब्दकोशमा प्रणालीमा प्रयोग गरिएका सबै डाटा तत्वहरूको बारेमा जानकारी समावेश हुनुपर्छ। यो जानकारीमा डेटा तत्व नामहरू, विवरणहरू, डेटा प्रकारहरू, लम्बाइहरू, ढाँचाहरू, र अन्य गुणहरू समावेश हुनुपर्छ।

क्याटलगिड: क्याटलगिड भनेको डाटा डिक्शनरीलाई व्यवस्थित गर्न र मर्मत गर्न सजिलो बनाउनको लागि तार्किक तरिकामा डेटा तत्वहरूलाई व्यवस्थित र समूहबद्ध गर्ने प्रक्रिया हो। यसमा डेटा तत्वहरूको कोटीहरू र उपश्रेणीहरू सिर्जना गर्न समावेश छ।

डाटा प्रक्रियाहरू: डाटा प्रक्रियाहरू प्रणालीमा डाटा सिर्जना, अद्यावधिक, पुनः प्राप्ति, वा मेटाउने चरणहरू हुन्। डाटा शब्दकोशमा सबै डाटा प्रक्रियाहरू र तिनीहरूसँग सम्बन्धित डाटा तत्वहरूको बारेमा जानकारी समावेश हुनुपर्छ।

डाटा प्रवाह: डाटा प्रवाह प्रणाली मार्फत डाटा को चाल को वर्णन गर्दछ। डाटा शब्दकोशमा सबै डाटा प्रवाहहरू र तिनीहरूसँग सम्बन्धित डाटा तत्वहरूको बारेमा जानकारी समावेश हुनुपर्छ।

डाटा स्टोर: डाटा स्टोर भनेको प्रणालीमा डाटा भण्डारण गरिएको ठाउँ हो। डाटा शब्दकोशमा सबै डाटा स्टोरहरू र तिनीहरूसँग सम्बन्धित डाटा तत्वहरूको बारेमा जानकारी समावेश हुनुपर्छ।

डाटा संरचना: डाटा संरचनाले डाटा स्टोर भित्र डाटा व्यवस्थित गर्ने तरिकालाई जनाउँछ। डाटा शब्दकोशले प्रणालीमा प्रयोग गरिएका सबै डाटा संरचनाहरूको बारेमा जानकारी समावेश गर्नुपर्छ।

डाटा तत्वहरू: डाटा तत्वहरू प्रणालीमा प्रयोग हुने जानकारीको आधारभूत एकाइहरू हुन्। डाटा शब्दकोशले प्रणालीमा प्रयोग गरिएका सबै डाटा तत्वहरूको बारेमा विस्तृत जानकारी समावेश गर्नुपर्छ।

डेटा शब्दकोशहरू संकलन गर्ने चरणहरू: डेटा शब्दकोश कम्पाइल गर्ने प्रक्रियामा निम्न चरणहरू समावेश छन्:

प्रणालीमा प्रयोग गरिएका सबै डाटा तत्वहरू पहिचान गर्नुहोस्

- तार्किक कोटिहरू र उपश्रेणीहरूमा डेटा तत्वहरूको सूची
- डेटा तत्वहरू र तिनीहरूका गुणहरू परिभाषित गर्नुहोस्, जस्तै डेटा प्रकार, लम्बाइ, ढाँचा, र अन्य विशेषताहरू
- कागजात डाटा प्रवाह, डाटा भण्डार, र डाटा संरचना
- डाटा शब्दकोशको शुद्धता प्रमाणित र प्रमाणित गर्नुहोस्
- प्रणालीमा परिवर्तनहरू गरिएपछि डेटा शब्दकोश अपडेट गर्नुहोस्।

यी चरणहरू पछ्याएर, एक व्यापक डेटा शब्दकोश सिर्जना गर्न सकिन्छ जसले प्रणालीमा प्रयोग गरिएको डाटाको विस्तृत बुझाइ प्रदान गर्दछ, जुन प्रणाली विश्लेषण, डिजाइन, र मर्मतका लागि उपयोगी हुन सक्छ।

Analyze structured decision system

- Information required for structured decision,
- Structured English
- Decision table and trees
- Completeness and accuracy

Information required for structured decision:

Structured decision-making involves using a defined process to make decisions based on clear and well-defined criteria. To analyze a structured decision system, it is essential to first identify the information that is required for the decision-making process. This information can be in the form of data, reports, documents, or any other relevant material. Understanding the information requirements is crucial to ensure that the decision-making process is efficient and effective.

Structured English:

Structured English is a method for specifying the steps required to solve a problem or perform a task in a structured decision system. It is a type of structured programming language that is used to describe the logic of the decision process in a clear and concise way. Structured English is designed to be easy to read and understand, making it a useful tool for communicating complex decision-making processes to stakeholders.

Decision table and trees:

Decision tables and trees are two common tools used in structured decision systems. A decision table is a matrix that outlines the criteria for a decision and the corresponding actions to be taken. It is useful for analyzing complex decision-making processes with multiple variables. Decision trees, on the other hand, are graphical representations of a decision-making process that map out all possible outcomes and decisions. Decision trees are useful for analyzing simple decision-making processes with a few variables.

Completeness and accuracy:

Analyzing the completeness and accuracy of a structured decision system is crucial to ensure that the system is reliable and effective. Completeness refers to the extent to which all possible scenarios have been considered in the decision-making process. Accuracy refers to the correctness of the decision-making process and the results it produces. Ensuring that the structured decision system is both complete and accurate is essential to ensure that the system meets the requirements of the stakeholders and produces reliable and trustworthy results.

संरचित निर्णय प्रणाली विश्लेषण गर्नुहोस्

- संरचित निर्णयको लागि आवश्यक जानकारी,
- संरचित अंग्रेजी

· निर्णय तालिका र रूखहरू

· पूर्णता र शुद्धता

संरचित निर्णयको लागि आवश्यक जानकारी:

संरचित निर्णय प्रक्रिया स्पष्ट र राम्ररी परिभाषित मापदण्डमा आधारित निर्णयहरू गर्न परिभाषित प्रक्रिया प्रयोग गर्न समावेश गर्दछ। संरचित निर्णय प्रणाली को विश्लेषण गर्न को लागी, यो पहिले निर्णय प्रक्रिया को लागी आवश्यक जानकारी को पहिचान गर्न आवश्यक छ। यो जानकारी डाटा, रिपोर्ट, कागजात, वा कुनै अन्य सान्दर्भिक सामग्री को रूप मा हुन सक्छ। निर्णय प्रक्रिया प्रभावकारी र प्रभावकारी छ भनी सुनिश्चित गर्न सूचना आवश्यकताहरू बुझ्न महत्वपूर्ण छ।

संरचित अंग्रेजी:

संरचित अंग्रेजी भनेको संरचित निर्णय प्रणालीमा समस्या समाधान गर्न वा कार्य गर्न आवश्यक चरणहरू निर्दिष्ट गर्ने विधि हो। यो एक प्रकारको संरचित प्रोग्रामिङ भाषा हो जुन निर्णय प्रक्रियाको तर्कलाई स्पष्ट र संक्षिप्त रूपमा वर्णन गर्न प्रयोग गरिन्छ। संरचित अङ्ग्रेजीलाई पढ्न र बुझ्न सजिलो हुने गरी डिजाइन गरिएको छ, यसले सरोकारवालाहरूलाई जटिल निर्णय प्रक्रियाहरू सञ्चार गर्न उपयोगी उपकरण बनाउँछ।

निर्णय तालिका र रूखहरू:

निर्णय तालिका र रूखहरू संरचित निर्णय प्रणालीहरूमा प्रयोग हुने दुई सामान्य उपकरणहरू हुन्। निर्णय तालिका एक म्याट्रिक्स हो जसले निर्णयको लागि मापदण्ड र लिनु पर्ने सम्बन्धित कार्यहरू रेखांकित गर्दछ। यो धेरै चरहरूको साथ जटिल निर्णय प्रक्रियाहरू विश्लेषण गर्न उपयोगी छ। अर्कोतर्फ, निर्णय रूखहरू निर्णय प्रक्रियाको ग्राफिकल प्रतिनिधित्व हुन् जसले सबै सम्भावित परिणामहरू र निर्णयहरूलाई नक्सा बनाउँछ। निर्णय रूखहरू केही चरहरूका साथ सरल निर्णय प्रक्रियाहरू विश्लेषण गर्न उपयोगी छन्।

पूर्णता र शुद्धता:

संरचित निर्णय प्रणालीको पूर्णता र शुद्धताको विश्लेषण प्रणाली भरपदो र प्रभावकारी छ भनी सुनिश्चित गर्न महत्वपूर्ण छ। पूर्णताले निर्णय गर्ने प्रक्रियामा सबै सम्भावित परिदृश्यहरूलाई विचार गरेको हदलाई जनाउँछ। शुद्धताले निर्णय लिने प्रक्रियाको शुद्धता र यसले उत्पादन गर्ने नतिजाहरूलाई जनाउँछ। संरचित निर्णय प्रणाली दुबै पूर्ण र सही छ भन्ने सुनिश्चित गर्नु आवश्यक छ कि प्रणालीले सरोकारवालाहरूको आवश्यकताहरू पूरा गर्दछ र भरपदो र विश्वसनीय परिणामहरू उत्पादन गर्दछ।

Analyze semi-structured decision system

- Risk in decision making,
- Types of decision; Analytic, Heuristic and Intelligence
- Design and choice

- Multiple criteria for decision making; Tradeoff, Weighting, Sequential elimination, and Goal programming

Analyzing semi-structured decision systems involves understanding the different types of decisions, designing decision-making processes, and considering multiple criteria for decision making. Here are some explanations of the topics:

Risk in decision making: This involves identifying the possible risks or uncertainties associated with a decision and evaluating their potential impact. Decision makers can use risk management techniques to assess and mitigate these risks.

Types of decision: There are three main types of decision-making processes: analytic, heuristic, and intelligence. Analytic decision making is based on a systematic approach that involves identifying and evaluating all possible alternatives. Heuristic decision making relies on rules of thumb, intuition, and past experiences. Intelligence decision making involves using creativity and insight to arrive at a solution.

Design and choice: This involve designing a decision-making process that is appropriate for the specific decision at hand. This can include identifying decision criteria, developing alternatives, and evaluating the alternatives against the criteria.

Multiple criteria for decision making: In some cases, there may be multiple criteria that need to be considered when making a decision. **Tradeoff** analysis involves evaluating the different criteria and making decisions that optimize the tradeoff between them. **Weighting** involves assigning weights to different criteria to reflect their relative importance. **Sequential elimination** involves evaluating the criteria one at a time and eliminating alternatives that do not meet each criterion. **Goal programming** involves identifying a set of goals and constraints and developing a decision-making process that achieves the goals while satisfying the constraints.

अर्ध-संरचित निर्णय प्रणाली विश्लेषण गर्नुहोस्

- निर्णय लिने जोखिम,
- निर्णयका प्रकारहरू; विश्लेषणात्मक, Heuristic र बुद्धिमत्ता
- डिजाइन र छनोट
- निर्णय लिनका लागि धेरै मापदण्डहरू; ट्रेडअफ, वजन, अनुक्रमिक उन्मूलन, र लक्ष्य प्रोग्रामिंग

अर्ध-संरचित निर्णय प्रणालीहरूको विश्लेषणमा विभिन्न प्रकारका निर्णयहरू बुझ्ने, निर्णय गर्ने प्रक्रियाहरू डिजाइन गर्ने, र निर्णय गर्ने धेरै मापदण्डहरू विचार गर्ने समावेश छ। यहाँ विषयहरूको केही व्याख्याहरू छन्:

निर्णय लिने जोखिम: यसले निर्णयसँग सम्बन्धित सम्भावित जोखिमहरू वा अनिश्चितताहरू पहिचान गर्ने र तिनीहरूको सम्भावित प्रभावहरूको मूल्याङ्कन समावेश गर्दछ। निर्णय निर्माताहरूले यी जोखिमहरू मूल्याङ्कन गर्ने र कम गर्ने जोखिम व्यवस्थापन प्रविधिहरू प्रयोग गर्न सक्छन्।

निर्णयका प्रकारहरू: त्यहाँ तीन मुख्य प्रकारका निर्णय गर्ने प्रक्रियाहरू छन्: विश्लेषणात्मक, हेरिस्टिक, र बुद्धि। विश्लेषणात्मक निर्णय एक व्यवस्थित दृष्टिकोणमा आधारित हुन्छ जसमा सबै सम्भावित विकल्पहरूको पहिचान र मूल्याङ्कन समावेश हुन्छ। ह्युरिस्टिक निर्णय लिने औँठा, अन्तर्ज्ञान, र विगतका अनुभवहरूको नियमहरूमा निर्भर हुन्छ। बुद्धिमत्ता निर्णय लिने एक समाधानमा पुग्न रचनात्मकता र अन्तरदृष्टि प्रयोग गर्ने समावेश छ।

डिजाइन र छनोट: यसमा हातमा रहेको विशिष्ट निर्णयको लागि उपयुक्त निर्णय गर्ने प्रक्रिया डिजाइन गर्ने समावेश छ। यसमा निर्णय मापदण्ड पहिचान गर्ने, विकल्पहरू विकास गर्ने, र मापदण्ड विरुद्ध विकल्पहरूको मूल्याङ्कन समावेश गर्न सकिन्छ।

निर्णय लिने धेरै मापदण्डहरू: केहि अवस्थामा, त्यहाँ धेरै मापदण्ड हुन सक्छ जुन निर्णय गर्दा विचार गर्ने आवश्यक छ। ट्रेडअफ विश्लेषणले विभिन्न मापदण्डहरूको मूल्याङ्कन गर्ने र तिनीहरू बीचको ट्रेडअफलाई अनुकूलन गर्ने निर्णयहरू समावेश गर्दछ। तौलमा फरक फरक मापदण्डहरूमा तौल तोक्ने समावेश छ ताकि तिनीहरूको सापेक्षिक महत्त्व प्रतिबिम्बित हुन्छ। क्रमिक उन्मूलनमा मापदण्डलाई एक पटकमा मूल्याङ्कन गर्ने र प्रत्येक मापदण्ड पूरा नगर्ने विकल्पहरू हटाउने समावेश हुन्छ। लक्ष्य प्रोग्रामिडले लक्ष्य र बाधाहरूको सेट पहिचान गर्ने र बाधाहरूलाई सन्तुष्ट गर्दै लक्ष्यहरू प्राप्त गर्ने निर्णय गर्ने प्रक्रियाको विकास समावेश गर्दछ।

Prepare/Present system proposal

- Inventorying and evaluation of computer hardware
- Workload estimation
- Acquisition of computer equipment, Vendor support
- Software evaluation
- Benefit/Cost, Payback, Cash flow analysis, trends
- Organising system proposal and its content

Preparing and presenting a system proposal involves several steps that help in evaluating the feasibility of the proposed system and presenting its benefits to stakeholders. These steps include:

Inventorying and evaluation of computer hardware: This involves taking stock of the existing computer hardware and evaluating its capacity and suitability for the proposed system.

Workload estimation: This involves estimating the expected workload on the proposed system to determine the appropriate hardware and software requirements.

Acquisition of computer equipment, Vendor support: This involves selecting vendors for the required computer equipment and negotiating support agreements with them.

Software evaluation: This involves evaluating the available software options and selecting the most appropriate one for the proposed system.

Benefit/Cost, Payback, Cash flow analysis, trends: This involves analyzing the financial feasibility of the proposed system by estimating its benefits and costs, calculating payback period and cash flow, and considering market trends.

Organizing system proposal and its content: This involves organizing the proposal into sections that cover the objectives, requirements, scope, methodology, timelines, resources, benefits, and costs of the proposed system.

Overall, the system proposal should provide a clear understanding of the proposed system and its benefits, including its impact on the organization's operations, customers, and stakeholders.

प्रणाली प्रस्ताव तयार/वर्तमान गर्नुहोस्

- कम्प्युटर हार्डवेयरको सूची र मूल्याङ्कन
- कार्यभार अनुमान
- कम्प्युटर उपकरणको अधिग्रहण, विक्रेता समर्थन
- सफ्टवेयर मूल्याङ्कन
- लाभ/लागत, भुक्तानी, नगद प्रवाह विश्लेषण, प्रवृत्ति
- व्यवस्थित प्रणाली प्रस्ताव र यसको सामग्री

प्रणाली प्रस्तावको तयारी र प्रस्तुतीकरणले प्रस्तावित प्रणालीको सम्भाव्यता मूल्याङ्कन गर्न र सरोकारवालाहरूलाई यसको फाइदाहरू प्रस्तुत गर्न मद्दत गर्ने धेरै चरणहरू समावेश गर्दछ। यी चरणहरू समावेश छन्:

कम्प्युटर हार्डवेयरको सूची र मूल्याङ्कन: यसमा अवस्थित कम्प्युटर हार्डवेयरको स्टक लिने र प्रस्तावित प्रणालीको लागि यसको क्षमता र उपयुक्तता मूल्याङ्कन समावेश छ।

कार्यभार अनुमान: यसमा उपयुक्त हार्डवेयर र सफ्टवेयर आवश्यकताहरू निर्धारण गर्न प्रस्तावित प्रणालीमा अपेक्षित कार्यभार अनुमान गर्न समावेश छ।

कम्प्युटर उपकरणको अधिग्रहण, विक्रेता समर्थन: यसमा आवश्यक कम्प्युटर उपकरणहरूको लागि विक्रेताहरू चयन गर्ने र तिनीहरूसँग समर्थन सम्झौताहरू वार्ता गर्ने समावेश छ।

सफ्टवेयर मूल्याङ्कन: यसमा उपलब्ध सफ्टवेयर विकल्पहरूको मूल्याङ्कन र प्रस्तावित प्रणालीको लागि सबैभन्दा उपयुक्त चयन समावेश छ।

लाभ/लागत, भुक्तानी, नगद प्रवाह विश्लेषण, प्रवृत्ति: यसमा यसको लाभ र लागत अनुमान गरेर, भुक्तानी अवधि र नगद प्रवाह गणना, र बजार प्रवृत्ति विचार गरेर प्रस्तावित प्रणालीको वित्तीय सम्भाव्यता विश्लेषण समावेश छ।

प्रणालीको प्रस्ताव र यसको सामग्रीलाई व्यवस्थित गर्ने: यसमा प्रस्तावित प्रणालीको उद्देश्य, आवश्यकता, दायरा, कार्यविधि, समयरेखा, स्रोत, लाभ र लागतहरू समावेश गर्ने खण्डहरूमा प्रस्तावलाई व्यवस्थित गर्ने समावेश छ।

समग्रमा, प्रणाली प्रस्तावले संस्थाको सञ्चालन, ग्राहकहरू र सरोकारवालाहरूमा पार्ने प्रभाव सहित प्रस्तावित प्रणाली र यसको फाइदाहरूको स्पष्ट बुझाइ प्रदान गर्नुपर्छ।

Module 5: Design essential

Design output

- Objectives of effective output
- Output technology
- Avoiding biases
- Functional and stylistic/Aesthetic attributes

Designing output is an essential part of any system design process. Here are some details on the different aspects of designing output:

Objectives of effective output: The main objective of designing effective output is to provide useful information to the users in a clear, concise, and understandable way. The output should be accurate, timely, relevant, and complete. The users should be able to easily interpret and use the information provided by the output.

Output technology: There are different output technologies available, such as paper-based output, electronic output, audio output, video output, etc. The technology used for output should be selected based on the user requirements, type of information to be presented, and the environment in which it will be used.

Avoiding biases: The output should be designed in a way that avoids biases, such as cultural, gender, age, or disability biases. The output should be designed to be accessible to all users, regardless of their background or abilities.

Functional and stylistic/aesthetic attributes: The output should have both functional and stylistic/aesthetic attributes. Functional attributes include the clarity and accuracy of the information provided, while stylistic/aesthetic attributes refer to the visual appeal of the output. The output should be designed in a way that is visually appealing and easy to read, while also conveying the necessary information.

Overall, designing effective output requires careful consideration of user requirements, output technology, and the functional and aesthetic attributes of the output. The output should be designed to provide useful information in a clear, concise, and understandable way, while also being accessible to all users and avoiding biases.

मोड्युल ५: डिजाइन आवश्यक

डिजाइन आउटपुट

- प्रभावकारी उत्पादनका उद्देश्यहरू
- आउटपुट प्रविधि
- पूर्वाग्रहबाट बच्ने
- कार्यात्मक र शैलीगत/सौंदर्य विशेषताहरू

डिजाइन आउटपुट कुनै पनि प्रणाली डिजाइन प्रक्रिया को एक आवश्यक भाग हो। यहाँ आउटपुट डिजाइन को विभिन्न पक्षहरू मा केहि विवरणहरू छन्:

प्रभावकारी आउटपुटका उद्देश्यहरू: प्रभावकारी आउटपुट डिजाइन गर्नुको मुख्य उद्देश्य प्रयोगकर्ताहरूलाई स्पष्ट, संक्षिप्त र बुझ्ने तरिकामा उपयोगी जानकारी प्रदान गर्नु हो। आउटपुट सही, समय सान्दर्भिक, सान्दर्भिक र पूर्ण हुनुपर्छ। प्रयोगकर्ताहरूले आउटपुट द्वारा प्रदान गरिएको जानकारी सजिलै व्याख्या गर्न र प्रयोग गर्न सक्षम हुनुपर्छ।

आउटपुट टेक्नोलोजी: कागजमा आधारित आउटपुट, इलेक्ट्रोनिक आउटपुट, अडियो आउटपुट, भिडियो आउटपुट, आदि जस्ता विभिन्न आउटपुट टेक्नोलोजीहरू उपलब्ध छन्। आउटपुटको लागि प्रयोग गरिने प्रविधि प्रयोगकर्ताको आवश्यकता, प्रस्तुत गरिने जानकारीको प्रकार, र प्रयोग गरिने वातावरणको आधारमा छनोट गर्नुपर्छ।

पूर्वाग्रहहरू बेवास्ता गर्दै: आउटपुटलाई पूर्वाग्रहहरू बेवास्ता गर्ने तरिकामा डिजाइन गरिनु पर्छ, जस्तै सांस्कृतिक, लिङ्ग, उमेर, वा असक्षमता पूर्वाग्रहहरू। आउटपुट सबै प्रयोगकर्ताहरूको लागि पहुँचयोग्य हुनको लागि डिजाइन गरिएको हुनुपर्छ, तिनीहरूको पृष्ठभूमि वा क्षमताहरू बिना।

कार्यात्मक र शैलीगत/सौंदर्यात्मक विशेषताहरू: आउटपुटमा कार्यात्मक र शैलीगत/सौंदर्यात्मक विशेषताहरू हुनुपर्छ। कार्यात्मक विशेषताहरूले प्रदान गरिएको जानकारीको स्पष्टता र शुद्धता समावेश गर्दछ, जबकि शैलीगत/सौंदर्यात्मक विशेषताहरूले आउटपुटको दृश्य अपीललाई जनाउँछ। आउटपुट आवश्यक जानकारी प्रदान गर्दै, दृश्यात्मक रूपमा आकर्षक र पढ्न सजिलो तरिकामा डिजाइन गरिएको हुनुपर्छ।

समग्रमा, प्रभावकारी आउटपुट डिजाइन गर्न प्रयोगकर्ता आवश्यकताहरू, आउटपुट टेक्नोलोजी, र आउटपुटको कार्यात्मक र सौंदर्य विशेषताहरूको सावधानीपूर्वक विचार गर्न आवश्यक छ। आउटपुटलाई स्पष्ट, संक्षिप्त र बुझ्ने तरिकामा उपयोगी जानकारी प्रदान गर्न डिजाइन गरिएको हुनुपर्छ, साथै सबै प्रयोगकर्ताहरूका लागि पहुँचयोग्य र पूर्वाग्रहहरूलाई बेवास्ता गर्दै।

Design input

- Guidelines for form design,
- Screen design; Icons and Colours

Design input refers to the process of designing the input methods for a computer system. This includes designing forms, screens, and other user interfaces that users interact with to input data into the system. The goal of design input is to create interfaces that are intuitive, efficient, and effective for users.

Guidelines for form design include creating forms that are easy to understand and use, with clear labels and instructions. Forms should be designed to minimize errors, with validation checks and error messages when necessary. The layout of the form should be organized logically, with related fields grouped together.

Screen design involves designing the visual layout of computer screens and the elements on those screens, such as buttons, menus, and text fields. **Icons and colours** can be used to help users navigate the system more easily and understand the meaning of different buttons and elements. For example, a red button might indicate that an action is critical or irreversible, while a green button might indicate that an action is safe and can be undone.

Overall, the goal of design input is to create interfaces that are easy to use and that help users input data accurately and efficiently. By following best practices and design guidelines, designers can create interfaces that are effective and meet the needs of users.

डिजाइन इनपुट

- फारम डिजाइनका लागि दिशानिर्देशहरू,
- पर्दा डिजाइन; प्रतीक र रंग

डिजाइन इनपुटले कम्प्युटर प्रणालीको लागि इनपुट विधिहरू डिजाइन गर्ने प्रक्रियालाई बुझाउँछ। यसमा डिजाइनिङ फारमहरू, स्क्रिनहरू, र अन्य प्रयोगकर्ता इन्टरफेसहरू समावेश छन् जुन प्रयोगकर्ताहरूले प्रणालीमा डेटा इनपुट गर्न अन्तरक्रिया गर्छन्। डिजाइन इनपुटको लक्ष्य प्रयोगकर्ताहरूको लागि सहज, कुशल र प्रभावकारी इन्टरफेसहरू सिर्जना गर्नु हो।

फारम डिजाइनका लागि दिशानिर्देशहरू स्पष्ट लेबल र निर्देशनहरू सहित बुझ्न र प्रयोग गर्न सजिलो फारमहरू सिर्जना गर्न समावेश गर्दछ। फारमहरू त्रुटिहरू कम गर्न डिजाइन गरिनु पर्छ, प्रमाणीकरण जाँचहरू र आवश्यक हुँदा त्रुटि सन्देशहरू सहित। फारमको लेआउट तार्किक रूपमा संगठित हुनुपर्छ, सम्बन्धित क्षेत्रहरू सँगै समूहबद्ध।

स्क्रिन डिजाइनले कम्प्युटर स्क्रिनको भिजुअल लेआउट र ती स्क्रिनहरूमा बटनहरू, मेनुहरू र पाठ क्षेत्रहरू जस्ता तत्वहरू डिजाइन गर्ने समावेश गर्दछ। आईकनहरू र रङहरू प्रयोगकर्ताहरूलाई प्रणालीमा अझ सजिलै नेभिगेट गर्न र विभिन्न बटन र तत्वहरूको अर्थ बुझ्न मद्दत गर्न प्रयोग गर्न सकिन्छ। उदाहरणका लागि, रातो बटनले कुनै कार्य महत्वपूर्ण वा अपरिवर्तनीय छ भनी संकेत गर्न सक्छ, जबकि हरियो बटनले कार्य सुरक्षित छ र त्यसलाई अन्डू गर्न सकिन्छ भनेर संकेत गर्न सक्छ।

समग्रमा, डिजाइन इनपुटको लक्ष्य प्रयोग गर्न सजिलो हुने इन्टरफेसहरू सिर्जना गर्नु हो र जसले प्रयोगकर्ताहरूलाई सही र प्रभावकारी रूपमा डेटा इनपुट गर्न मद्दत गर्दछ। उत्कृष्ट अभ्यासहरू र डिजाइन

दिशानिर्देशहरू पछ्याएर, डिजाइनहरूले प्रभावकारी र प्रयोगकर्ताहरूको आवश्यकताहरू पूरा गर्ने इन्टरफेसहरू सिर्जना गर्न सक्छन्।

Design file/database

- Conventional files and databases
- Data concepts
- File and database organisation
- Normalisation

Designing a file/database involves creating an organized structure to store and retrieve data efficiently. The following are the key aspects of designing a file/database:

Conventional files and databases: Conventional files are traditional storage systems used to store and retrieve data. They are usually created using tools like Microsoft Excel or Google Sheets. In contrast, databases are more advanced and use a structured approach to store and manage data. They are designed to efficiently store, retrieve, and manage large amounts of data. Databases can be relational, hierarchical, or object-oriented.

Data concepts: Data concepts are the building blocks of a database. They include entities, attributes, and relationships. Entities are objects or concepts that are represented in a database, while attributes are the characteristics of those entities. Relationships describe how entities are related to one another.

File and database organisation: File and database organisation refers to the way data is stored and retrieved from a file or database. It includes file structures, indexing methods, and access methods. File structures determine how data is organised within a file, while indexing methods and access methods provide efficient ways of locating and retrieving data.

Normalisation: Normalisation is the process of organising data in a database to eliminate redundancy and improve data integrity. It involves breaking down a database into smaller, more manageable tables and establishing relationships between those tables. Normalisation ensures that data is stored in a consistent and efficient manner, reducing the risk of errors and inconsistencies in the data.

Design user interface

- Types of user interface; Natural language, Question-and-Answer, Menus, input/output, Command language, Direct manipulation and the Mouse

User interface design refers to the process of designing interfaces in software, applications, or devices that facilitate interaction between users and the system. It involves the arrangement of graphical elements, input controls, navigation mechanisms, and other features to enable efficient and intuitive communication with the system.

Here are the different types of user interface:

Natural language interface: This type of interface allows users to interact with the system using natural language, such as spoken or written commands. The system interprets the input and provides responses in a human-like manner.

Question-and-answer interface: In this type of interface, the system asks users a series of questions to elicit the information it needs to complete a task. The users respond with answers, and the system processes the responses to produce the desired output.

Menus interface: This interface provides users with a set of options or commands displayed in a hierarchical menu structure. Users select options from the menu to initiate actions or navigate through the system.

Input/output interface: This interface allows users to input data into the system and receive output from it. Examples include text boxes, drop-down menus, and radio buttons.

Command language interface: This interface requires users to type in commands in a specific syntax to initiate actions or manipulate data. It is commonly used in command-line interfaces for operating systems or programming languages.

Direct manipulation interface: This type of interface allows users to manipulate graphical objects directly using gestures or movements. Examples include drag-and-drop, pinch-to-zoom, and swipe gestures.

Mouse interface: This interface involves using a mouse or other pointing device to navigate and interact with the system. It is commonly used in graphical user interfaces (GUIs) for desktop applications.

Each type of user interface has its advantages and disadvantages depending on the context of use, user needs, and system requirements. Therefore, it is essential to consider these factors when designing a user interface to ensure its effectiveness and usability.

डिजाइन फाइल/डाटाबेस

- परम्परागत फाइलहरू र डाटाबेसहरू
- डाटा अवधारणाहरू
- फाइल र डाटाबेस संगठन
- सामान्यीकरण

फाइल/डेटाबेस डिजाइन गर्दा डाटा कुशलतापूर्वक भण्डारण र पुनःप्राप्त गर्न संगठित संरचना सिर्जना गर्नु समावेश छ। फाइल/डेटाबेस डिजाइन गर्ने मुख्य पक्षहरू निम्न हुन्:

परम्परागत फाइलहरू र डाटाबेसहरू: परम्परागत फाइलहरू डेटा भण्डारण र पुनः प्राप्त गर्न प्रयोग गरिने परम्परागत भण्डारण प्रणालीहरू हुन्। तिनीहरू सामान्यतया Microsoft Excel वा Google पानाहरू जस्तै उपकरणहरू प्रयोग गरेर सिर्जना गरिन्छ। यसको विपरित, डाटाबेसहरू अधिक उन्नत छन् र डाटा भण्डारण र

व्यवस्थापन गर्न एक संरचित दृष्टिकोण प्रयोग गर्दछ। तिनीहरू कुशलतापूर्वक भण्डारण गर्न, पुनः प्राप्त गर्न, र ठूलो मात्रामा डेटा व्यवस्थापन गर्न डिजाइन गरिएको हो। डाटाबेस रिलेशनल, पदानुक्रमिक, वा वस्तु-उन्मुख हुन सक्छ।

डाटा अवधारणाहरू: डाटा अवधारणाहरू डाटाबेसको निर्माण ब्लकहरू हुन्। तिनीहरूमा संस्थाहरू, विशेषताहरू, र सम्बन्धहरू समावेश छन्। संस्थाहरू वस्तु वा अवधारणाहरू हुन् जुन डेटाबेसमा प्रतिनिधित्व गरिन्छ, जबकि विशेषताहरू ती संस्थाहरूको विशेषताहरू हुन्। सम्बन्धहरूले एकअर्कासँग कसरी संस्थाहरू सम्बन्धित छन् भनेर वर्णन गर्दछ।

फाइल र डाटाबेस संगठन: फाइल र डाटाबेस संगठनले फाइल वा डाटाबेसबाट डाटा भण्डारण र पुनःप्राप्त गर्ने तरिकालाई जनाउँछ। यसले फाइल संरचनाहरू, अनुक्रमणिका विधिहरू, र पहुँच विधिहरू समावेश गर्दछ। फाइल ढाँचाहरूले फाइल भित्र डाटा कसरी व्यवस्थित गरिन्छ भनेर निर्धारण गर्दछ, जबकि अनुक्रमणिका विधिहरू र पहुँच विधिहरूले डाटा पत्ता लगाउने र पुनः प्राप्त गर्ने प्रभावकारी तरिकाहरू प्रदान गर्दछ।

सामान्यीकरण: सामान्यीकरण भनेको अनावश्यकता हटाउन र डाटा अखण्डता सुधार गर्न डाटाबेसमा डाटा व्यवस्थित गर्ने प्रक्रिया हो। यसले डाटाबेसलाई साना, अधिक व्यवस्थित तालिकाहरूमा तोड्ने र ती तालिकाहरू बीच सम्बन्ध स्थापना गर्ने समावेश गर्दछ। सामान्यीकरणले डेटामा त्रुटि र असंगतताहरूको जोखिम कम गर्दै, एक सुसंगत र प्रभावकारी तरिकामा डाटा भण्डारण गरिएको सुनिश्चित गर्दछ।

डिजाइन प्रयोगकर्ता इन्टरफेस

प्रयोगकर्ता इन्टरफेस को प्रकार; प्राकृतिक भाषा, प्रश्न-उत्तर, मेनु, इनपुट/आउटपुट, आदेश भाषा, प्रत्यक्ष हेरफेर र माउस

प्रयोगकर्ता इन्टरफेस डिजाइनले सफ्टवेयर, अनुप्रयोगहरू, वा उपकरणहरूमा इन्टरफेसहरू डिजाइन गर्ने प्रक्रियालाई जनाउँछ जसले प्रयोगकर्ताहरू र प्रणाली बीचको अन्तरक्रियालाई सहज बनाउँछ। यसमा ग्राफिकल तत्वहरू, इनपुट नियन्त्रणहरू, नेभिगेसन संयन्त्रहरू, र प्रणालीसँग कुशल र सहज संचार सक्षम गर्न अन्य सुविधाहरूको व्यवस्था समावेश छ।

यहाँ प्रयोगकर्ता इन्टरफेस को विभिन्न प्रकार छन्:

प्राकृतिक भाषा इन्टरफेस: यस प्रकारको इन्टरफेसले प्रयोगकर्ताहरूलाई प्राकृतिक भाषा, जस्तै बोल्ने वा लिखित आदेशहरू प्रयोग गरेर प्रणालीसँग अन्तरक्रिया गर्न अनुमति दिन्छ। प्रणालीले इनपुटको व्याख्या गर्छ र मानव-जस्तै प्रतिक्रियाहरू प्रदान गर्दछ।

प्रश्न-उत्तर इन्टरफेस: यस प्रकारको इन्टरफेसमा, प्रणालीले प्रयोगकर्ताहरूलाई कार्य पूरा गर्न आवश्यक जानकारी प्राप्त गर्न प्रश्नहरूको श्रृंखला सोध्छ। प्रयोगकर्ताहरूले जवाफहरूको साथ प्रतिक्रिया दिन्छन्, र प्रणालीले इच्छित आउटपुट उत्पादन गर्न प्रतिक्रियाहरू प्रशोधन गर्दछ।

मेनु इन्टरफेस: यो इन्टरफेसले प्रयोगकर्ताहरूलाई श्रेणीबद्ध मेनु संरचनामा देखाइएका विकल्पहरू वा आदेशहरूको सेट प्रदान गर्दछ। प्रयोगकर्ताहरूले कार्यहरू सुरु गर्न वा प्रणाली मार्फत नेभिगेट गर्न मेनुबाट विकल्पहरू चयन गर्छन्।

इनपुट/आउटपुट इन्टरफेस: यो इन्टरफेसले प्रयोगकर्ताहरूलाई प्रणालीमा डाटा इनपुट गर्न र त्यसबाट आउटपुट प्राप्त गर्न अनुमति दिन्छ। उदाहरणहरूमा पाठ बाक्सहरू, ड्रप-डाउन मेनुहरू, र रेडियो बटनहरू समावेश छन्।

आदेश भाषा इन्टरफेस: यो इन्टरफेसले प्रयोगकर्ताहरूलाई कार्यहरू प्रारम्भ गर्न वा डेटा हेरफेर गर्न विशेष वाक्य रचनामा आदेशहरू टाइप गर्न आवश्यक छ। यो सामान्यतया अपरेटिङ सिस्टम वा प्रोग्रामिङ भाषाहरूको लागि कमाण्ड-लाइन इन्टरफेसहरूमा प्रयोग गरिन्छ।

प्रत्यक्ष हेरफेर इन्टरफेस: यस प्रकारको इन्टरफेसले प्रयोगकर्ताहरूलाई इशारा वा चालहरू प्रयोग गरेर प्रत्यक्ष रूपमा ग्राफिकल वस्तुहरू हेरफेर गर्न अनुमति दिन्छ। उदाहरणहरूमा ड्र्याग-एन्ड-ड्रप, पिन्च-टु-जुम, र स्वाइप इशाराहरू समावेश छन्।

माउस इन्टरफेस: यो इन्टरफेसले नेभिगेट गर्न र प्रणालीसँग अन्तरक्रिया गर्न माउस वा अन्य पोइन्टिङ उपकरण प्रयोग गर्दछ। यो सामान्यतया डेस्कटप अनुप्रयोगहरूको लागि ग्राफिकल प्रयोगकर्ता इन्टरफेस (GUIs) मा प्रयोग गरिन्छ।

प्रत्येक प्रकारको प्रयोगकर्ता इन्टरफेससँग यसको फाइदा र बेफाइदाहरू छन् प्रयोगको सन्दर्भमा निर्भर गर्दछ, प्रयोगकर्ता आवश्यकताहरू, र प्रणाली आवश्यकताहरू। त्यसकारण, यसको प्रभावकारिता र उपयोगिता सुनिश्चित गर्न प्रयोगकर्ता इन्टरफेस डिजाइन गर्दा यी कारकहरूलाई विचार गर्न आवश्यक छ।

Design data entry procedures

- Objectives
- Purpose of coding
- Effective/Accurate coding
- Bottleneck in data entry

Design data entry procedures involves planning and designing the process of how data will be entered into a computer system. There are four important aspects of designing data entry procedures:

Objectives: The objectives are the goals that we want to achieve by entering the data into the system. For example, we might want to track inventory levels or customer information.

Purpose of coding: Coding is the process of assigning a code to each piece of data so that the computer can easily recognize and process it. The purpose of coding is to make data entry more efficient and accurate.

Effective/Accurate coding: Effective coding means that the codes we use accurately reflect the data we are entering. For example, if we are coding customer names, we would use a code that

is unique to each customer. Accurate coding means that the codes are entered correctly into the system.

Bottleneck in data entry: A bottleneck is a point in the data entry process where the speed or efficiency slows down. For example, if the computer system is slow, it could cause a bottleneck in the data entry process. Other bottlenecks could include errors in coding or data entry, or a lack of training for data entry staff.

By designing effective data entry procedures, we can ensure that the data entered the system is accurate, efficient, and useful for achieving our objectives.

डाटा प्रविष्टि प्रक्रियाहरू डिजाइन गर्नुहोस्

- उद्देश्य
- कोडिङको उद्देश्य
- प्रभावकारी/सटीक कोडिङ
- डाटा प्रविष्टिमा अवरोध

डिजाइन डाटा प्रविष्टि प्रक्रियाहरूले कम्प्युटर प्रणालीमा कसरी डाटा प्रविष्टि गर्ने प्रक्रियाको योजना र डिजाइन समावेश गर्दछ। डाटा प्रविष्टि प्रक्रियाहरू डिजाइन गर्ने चार महत्त्वपूर्ण पक्षहरू छन्:

उद्देश्यहरू: उद्देश्यहरू लक्ष्यहरू हुन् जुन हामीले प्रणालीमा डाटा प्रविष्टि गरेर प्राप्त गर्न चाहन्छौं। उदाहरणका लागि, हामी सूची स्तर वा ग्राहक जानकारी ड्र्याक गर्न चाहन्छौं।

कोडिङको उद्देश्य: कोडिङ भनेको डाटाको प्रत्येक टुकामा कोड प्रदान गर्ने प्रक्रिया हो ताकि कम्प्युटरले सजिलै चिन्न र प्रशोधन गर्न सकोस्। कोडिङको उद्देश्य डाटा प्रविष्टिलाई अझ प्रभावकारी र सटीक बनाउनु हो।

प्रभावकारी/सटीक कोडिङ: प्रभावकारी कोडिङ भनेको हामीले प्रयोग गर्ने कोडहरू हामीले प्रविष्टि गर्ने डाटालाई सही रूपमा प्रतिबिम्बित गर्ने हो। उदाहरणका लागि, यदि हामी ग्राहकको नामहरू कोडिङ गर्दैछौं भने, हामी प्रत्येक ग्राहकको लागि अद्वितीय कोड प्रयोग गर्नेछौं। सही कोडिङ भनेको प्रणालीमा कोडहरू सही रूपमा प्रविष्टि गरिएको हो।

डेटा प्रविष्टिमा अवरोध: एक अवरोध डेटा प्रविष्टि प्रक्रियामा एक बिन्दु हो जहाँ गति वा दक्षता कम हुन्छ। उदाहरणका लागि, यदि कम्प्युटर प्रणाली ढिलो छ भने, यसले डाटा प्रविष्टि प्रक्रियामा बाधा उत्पन्न गर्न सक्छ। अन्य अवरोधहरूमा कोडिङ वा डाटा प्रविष्टिमा त्रुटिहरू, वा डाटा प्रविष्टि कर्मचारीहरूको लागि प्रशिक्षणको कमी समावेश हुन सक्छ।

प्रभावकारी डेटा प्रविष्टि प्रक्रियाहरू डिजाइन गरेर, हामी सुनिश्चित गर्न सक्छौं कि प्रणालीमा प्रविष्टि गरिएको डाटा हाम्रो उद्देश्यहरू प्राप्त गर्नको लागि सही, कुशल र उपयोगी छ।

Deploy project

◆ Implementation approaches

When it comes to deploying a project, there are several implementation approaches that can be taken, including:

Direct cutover: In this approach, the new system replaces the old one all at once. This can be risky, as there is no backup if the new system fails.

Phased implementation: In this approach, the new system is implemented in stages, one module or department at a time. This reduces risk and allows for easier testing and debugging.

Parallel implementation: In this approach, the new system is implemented alongside the old one for a period. This allows for testing and comparison of the two systems, but can be more costly and time-consuming.

Pilot implementation: In this approach, the new system is implemented in a small, controlled setting to test its effectiveness before being rolled out to the entire organization.

Once an implementation approach has been chosen, the deployment process can begin. This involves installing the new system, testing it, and training users on how to use it. It may also involve migrating data from the old system to the new one. After deployment, ongoing support and maintenance are important to ensure the system continues to function effectively.

परियोजना लागू गर्नुहोस्

· कार्यान्वयन दृष्टिकोणहरू

जब यो एक परियोजना को तैनाती को लागी आउँछ, त्यहाँ धेरै कार्यान्वयन दृष्टिकोणहरू छन् जुन लिन सकिन्छ, सहित:

प्रत्यक्ष कटओभर: यस दृष्टिकोणमा, नयाँ प्रणालीले पुरानोलाई एकैचोटि प्रतिस्थापन गर्छ। यो जोखिमपूर्ण हुन सक्छ, किनकि नयाँ प्रणाली असफल भएमा त्यहाँ कुनै ब्याकअप छैन।

चरणबद्ध कार्यान्वयन: यस दृष्टिकोणमा, नयाँ प्रणाली चरणहरूमा, एक मोड्युल वा विभागमा एक पटकमा लागू गरिन्छ। यसले जोखिम घटाउँछ र सजिलो परीक्षण र डिबगिङको लागि अनुमति दिन्छ।

समानान्तर कार्यान्वयन: यस दृष्टिकोणमा, नयाँ प्रणाली एक अवधिको लागि पुरानो प्रणालीसँगै लागू हुन्छ। यसले दुई प्रणालीहरूको परीक्षण र तुलनाको लागि अनुमति दिन्छ, तर अधिक महँगो र समय-उपभोग हुन सक्छ।

पायलट कार्यान्वयन: यस दृष्टिकोणमा, नयाँ प्रणालीलाई सम्पूर्ण संगठनमा रोल आउट गर्नु अघि यसको प्रभावकारिता परीक्षण गर्न सानो, नियन्त्रित सेटिङमा लागू गरिन्छ।

एक पटक कार्यान्वयन दृष्टिकोण छनोट भएपछि, तैनाती प्रक्रिया सुरु गर्न सकिन्छ। यसमा नयाँ प्रणाली स्थापना गर्ने, परीक्षण गर्ने र प्रयोगकर्ताहरूलाई यसलाई कसरी प्रयोग गर्ने भनेर प्रशिक्षण दिने काम समावेश छ। यसमा पुरानो प्रणालीबाट नयाँ प्रणालीमा डाटा माइग्रेट गर्ने पनि समावेश हुन सक्छ। तैनाती पछि, प्रणालीले प्रभावकारी रूपमा काम गर्न जारी राखेको सुनिश्चित गर्न निरन्तर समर्थन र मर्मत महत्वपूर्ण छ।

Module 6: Control quality of system

- Explain total quality assurance approach
- Quality assurance approaches, Structured walkthrough
- Bottom-up and Topdown design

Total quality assurance approach:

The total quality assurance approach is a method used to ensure that a system meets the requirements and specifications of the stakeholders involved in the project. This approach involves using a range of techniques and tools to ensure that the system is tested thoroughly and meets the required quality standards.

Quality assurance approaches:

There are several quality assurance approaches that can be used to test a system, including structured walkthroughs. A structured walkthrough is a method of reviewing a system where a group of people examine the system to find any issues or potential problems. This is an effective way to ensure that the system meets the required standards and can identify any issues early in the development process.

Bottom-up and top-down design:

Bottom-up and top-down design are two approaches used to develop a system. Bottom-up design involves developing individual components of the system and then combining them to form the final product. Top-down design involves starting with the overall system design and then breaking it down into smaller components. Both approaches have their advantages and disadvantages, and the choice of which approach to use will depend on the specific project requirements and constraints.

मोड्युल 6: प्रणालीको गुणस्तर नियन्त्रण

- कुल गुणस्तर आश्वासन दृष्टिकोण व्याख्या गर्नुहोस्
- गुणस्तर आश्वासन दृष्टिकोण, संरचित वाकथु
- बटम-अप र टपडाउन डिजाइन

कुल गुणस्तर आश्वासन दृष्टिकोण:

कुल गुणस्तर आश्वासन दृष्टिकोण भनेको प्रणालीले परियोजनामा संलग्न सरोकारवालाहरूको आवश्यकता र विशिष्टताहरू पूरा गर्छ भनी सुनिश्चित गर्न प्रयोग गरिने विधि हो। यस दृष्टिकोणले प्रणालीलाई राम्ररी परीक्षण गरिएको छ र आवश्यक गुणस्तर मापदण्डहरू पूरा गरेको छ भनी सुनिश्चित गर्न विभिन्न प्रविधि र उपकरणहरूको प्रयोग समावेश गर्दछ।

गुणस्तर आश्वासन दृष्टिकोण:

त्यहाँ धेरै गुणस्तर आश्वासन दृष्टिकोणहरू छन् जुन संरचित वाक्यहरू सहित प्रणाली परीक्षण गर्न प्रयोग गर्न सकिन्छ। एक संरचित वाक्य एक प्रणालीको समीक्षा गर्ने एक विधि हो जहाँ मानिसहरूको समूहले कुनै पनि समस्या वा सम्भावित समस्याहरू फेला पार्ने प्रणालीको जाँच गर्दछ। यो प्रणालीले आवश्यक मापदण्डहरू पूरा गर्छ र विकास प्रक्रियाको प्रारम्भमा कुनै पनि समस्याहरू पहिचान गर्न सक्छ भनी सुनिश्चित गर्न यो प्रभावकारी तरिका हो।

तल-माथि र माथि-डाउन डिजाइन:

तल्लो-माथि र माथि-डाउन डिजाइन प्रणाली विकास गर्न प्रयोग गरिने दुई दृष्टिकोणहरू हुन्। बटम-अप डिजाइनले प्रणालीको व्यक्तिगत कम्पोनेन्टहरू विकास गर्ने र त्यसपछि तिनीहरूलाई अन्तिम उत्पादन बनाउन संयोजन गर्दछ। शीर्ष-डाउन डिजाइनले समग्र प्रणाली डिजाइनको साथ सुरु गरी यसलाई साना भागहरूमा तोड्ने समावेश गर्दछ। दुबै दृष्टिकोणका आ-आफ्ना फाइदा र बेफाइदाहरू छन्, र कुन दृष्टिकोण प्रयोग गर्ने छनोट विशेष परियोजना आवश्यकताहरू र अवरोधहरूमा निर्भर हुनेछ।

Experiment system

- Testing processes
- White Box and Black Box test
- Bottom-up and Topdown test

Experimenting a system involves various testing processes to ensure that it meets the specified requirements and is functioning correctly.

Testing is an essential part of developing any system to ensure it is functioning correctly and meets the intended requirements. Here are some of the key aspects of testing processes:

Testing processes: There are several steps involved in the testing process, such as defining test cases, executing them, and reporting the results. It's important to create a comprehensive test plan and ensure that all aspects of the system are tested thoroughly to identify any potential issues.

White Box and Black Box test: These are two main approaches to testing a system. White box testing involves examining the internal structure of the system, including the code, to test its functionality. Black box testing, on the other hand, focuses on the external behavior of the system and tests its functionality without looking at the code.

Bottom-up and Top-down test: These are two main testing strategies. Bottom-up testing starts with testing individual components of the system and then testing their integration with each

other to ensure that the whole system is functioning correctly. Top-down testing, on the other hand, starts with testing the overall system and then testing individual components to ensure that they are working correctly.

प्रयोग प्रणाली

- परीक्षण प्रक्रियाहरू
- सेतो बक्स र कालो बक्स परीक्षण
- बटम-अप र टपडाउन परीक्षण

प्रणाली प्रयोग गर्दा यसले निर्दिष्ट आवश्यकताहरू पूरा गर्दछ र सही रूपमा काम गरिरहेको छ भनी सुनिश्चित गर्न विभिन्न परीक्षण प्रक्रियाहरू समावेश गर्दछ।

परीक्षण कुनै पनि प्रणालीलाई सही ढंगले काम गरिरहेको छ र अपेक्षित आवश्यकताहरू पूरा गर्दछ भनी सुनिश्चित गर्नको लागि विकासको एक आवश्यक भाग हो। यहाँ परीक्षण प्रक्रियाका केही प्रमुख पक्षहरू छन्:

परीक्षण प्रक्रियाहरू: त्यहाँ परीक्षण प्रक्रियामा धेरै चरणहरू समावेश छन्, जस्तै परीक्षण केसहरू परिभाषित गर्ने, तिनीहरूलाई कार्यान्वयन गर्ने, र परिणामहरू रिपोर्ट गर्ने। यो एक व्यापक परीक्षण योजना सिर्जना गर्न र कुनै पनि सम्भावित समस्याहरू पहिचान गर्न प्रणालीका सबै पक्षहरू राम्ररी परीक्षण गरिएको छ भनी सुनिश्चित गर्न महत्वपूर्ण छ।

सेतो बक्स र ब्ल्याक बक्स परीक्षण: यी प्रणाली परीक्षण गर्न दुई मुख्य दृष्टिकोणहरू हुन्। सेतो बक्स परीक्षणले यसको कार्यक्षमता परीक्षण गर्न कोड सहित प्रणालीको आन्तरिक संरचनाको परीक्षण समावेश गर्दछ। ब्ल्याक बक्स परीक्षण, अर्कोतर्फ, प्रणालीको बाह्य व्यवहारमा ध्यान केन्द्रित गर्दछ र कोड नहेरी यसको कार्यक्षमता परीक्षण गर्दछ।

बटम-अप र टप-डाउन परीक्षण: यी दुई मुख्य परीक्षण रणनीतिहरू हुन्। बटम-अप परीक्षण प्रणालीको व्यक्तिगत कम्पोनेन्टहरूको परीक्षणको साथ सुरु हुन्छ र त्यसपछि सम्पूर्ण प्रणाली सही रूपमा काम गरिरहेको छ भनी सुनिश्चित गर्न एक अर्कासँग तिनीहरूको एकीकरण परीक्षण गरेर। शीर्ष-डाउन परीक्षण, अर्कोतर्फ, समग्र प्रणालीको परीक्षणको साथ सुरु हुन्छ र त्यसपछि तिनीहरूले सही रूपमा काम गर्दै छन् भनेर सुनिश्चित गर्न व्यक्तिगत घटकहरूको परीक्षण गर्दछ।

Maintain system

- Maintenance practices

System maintenance is an important process that involves keeping a computer system running efficiently and effectively. The goal of maintenance is to ensure that the system operates correctly, and that all components are updated, repaired, or replaced as necessary to prevent system downtime or failures.

Maintenance practices include a range of activities that are performed regularly to keep the system running smoothly. These may include:

Preventive Maintenance: This includes tasks such as checking for software and hardware updates, replacing worn-out components, cleaning the system, and conducting regular backups.

Corrective Maintenance: This includes repairing or replacing faulty hardware or software components and fixing any system bugs or errors.

Adaptive Maintenance: This involves modifying or enhancing the system to meet changing business requirements, user needs, or technology advancements.

Perfective Maintenance: This includes improving the system's performance, scalability, or usability by adding new features or functionality.

Emergency Maintenance: This is performed when a system failure occurs, and it is needed to restore the system to a functional state.

By following good maintenance practices, a system can remain in good working order for an extended period. This can help prevent system downtime, reduce repair costs, and extend the useful life of the system.

प्रणाली कायम गर्ने

• मर्मत अभ्यास

प्रणाली मर्मत एक महत्वपूर्ण प्रक्रिया हो जसमा कम्प्युटर प्रणालीलाई प्रभावकारी र प्रभावकारी रूपमा चलाउन समावेश छ। मर्मतसम्भारको लक्ष्य भनेको प्रणाली सही रूपमा सञ्चालन हुन्छ, र प्रणाली डाउनटाइम वा विफलताहरू रोक्नको लागि आवश्यक रूपमा सबै कम्पोनेन्टहरू अद्यावधिक, मर्मत, वा प्रतिस्थापन गरिएको सुनिश्चित गर्नु हो।

मर्मत अभ्यासहरूले प्रणालीलाई सुचारु रूपमा चलाउनको लागि नियमित रूपमा प्रदर्शन गर्ने गतिविधिहरूको दायरा समावेश गर्दछ। यसमा समावेश हुन सक्छ:

निवारक मर्मतसम्भार: यसमा सफ्टवेयर र हार्डवेयर अपडेटहरूको लागि जाँच गर्ने, थकित कम्पोनेन्टहरू बदल्ने, प्रणाली सफा गर्ने, र नियमित ब्याकअपहरू सञ्चालन गर्ने जस्ता कार्यहरू समावेश छन्।

सुधारात्मक मर्मत: यसमा दोषपूर्ण हार्डवेयर वा सफ्टवेयर कम्पोनेन्टहरू मर्मत गर्ने वा प्रतिस्थापन गर्ने र कुनै पनि प्रणाली बगहरू वा त्रुटिहरू समाधान गर्ने समावेश छ।

अनुकूलनीय मर्मतसम्भार: यसले परिवर्तन गर्ने व्यापार आवश्यकताहरू, प्रयोगकर्ता आवश्यकताहरू, वा प्रविधिको उन्नतिहरू पूरा गर्ने प्रणालीलाई परिमार्जन वा विस्तार समावेश गर्दछ।

पूर्ण मर्मतसम्भार: यसले नयाँ सुविधाहरू वा कार्यक्षमता थपेर प्रणालीको कार्यसम्पादन, स्केलेबिलिटी, वा उपयोगिता सुधार गर्ने समावेश गर्दछ।

आपतकालीन मर्मत: यो प्रदर्शन गरिन्छ जब प्रणाली विफलता हुन्छ, र यो प्रणालीलाई कार्यात्मक अवस्थामा पुनर्स्थापित गर्ने आवश्यक छ।

राम्रो मर्मत अभ्यासहरू पालना गरेर, प्रणाली विस्तारित अवधिको लागि राम्रो कार्य क्रममा रहन सक्छ। यसले प्रणाली डाउनटाइम रोक्न, मर्मत लागत घटाउन र प्रणालीको उपयोगी जीवन विस्तार गर्न मद्दत गर्न सक्छ।

Audit system

◆ Internal/External auditing

An audit is a systematic and independent examination of a system, process, or organization to determine whether the established procedures and policies are being followed and to identify areas that require improvement. In the context of information systems, an audit is performed to evaluate the effectiveness and efficiency of an information system, to ensure compliance with laws and regulations, and to identify risks and vulnerabilities.

Internal auditing involves evaluating the information system by an internal audit department or unit within an organization. The primary objective of internal auditing is to provide assurance to the management that the information system is functioning effectively and efficiently and follows laws, regulations, and policies. Internal auditing also aims to identify risks and vulnerabilities and suggest recommendations for improvements.

External auditing involves the evaluation of the information system by an independent auditor who is not a part of the organization. The primary objective of external auditing is to provide an unbiased opinion about the effectiveness and efficiency of the information system. External auditors typically perform audits to ensure compliance with laws and regulations, identify risks and vulnerabilities, and suggest recommendations for improvement. The external audit report is usually submitted to the management and other stakeholders, including regulatory authorities, investors, and customers.

Both **internal and external auditing** are essential for ensuring the quality and integrity of the information system. The audit process typically involves reviewing documentation, interviewing personnel, and testing system controls and procedures. The audit findings and recommendations are then reported to the management, and corrective actions are taken to address any issues identified during the audit.

लेखापरीक्षण प्रणाली

· आन्तरिक/बाह्य लेखापरीक्षण

लेखापरीक्षण प्रणाली, प्रक्रिया वा संस्थाको व्यवस्थित र स्वतन्त्र परीक्षण हो जुन स्थापित प्रक्रिया र नीतिहरू पछ्याइएको छ कि छैन भनेर निर्धारण गर्न र सुधार आवश्यक पर्ने क्षेत्रहरू पहिचान गर्न। सूचना प्रणालीको सन्दर्भमा, सूचना प्रणालीको प्रभावकारिता र दक्षता मूल्याङ्कन गर्न, कानून र नियमहरूको पालना सुनिश्चित गर्न, र जोखिम र कमजोरीहरू पहिचान गर्न लेखापरीक्षण गरिन्छ।

आन्तरिक लेखापरीक्षणले संगठन भित्रको आन्तरिक लेखापरीक्षण विभाग वा इकाईद्वारा सूचना प्रणालीको मूल्याङ्कन समावेश गर्दछ। आन्तरिक लेखापरीक्षणको मुख्य उद्देश्य भनेको व्यवस्थापनलाई सूचना प्रणालीले प्रभावकारी र प्रभावकारी रूपमा काम गरिरहेको छ र कानून, नियम र नीतिहरूको पालना गरेको छ भन्ने आश्वासन प्रदान गर्नु हो। आन्तरिक लेखापरीक्षणले जोखिम र कमजोरीहरू पहिचान गर्ने र सुधारका लागि सिफारिसहरू सुझाव दिने पनि लक्ष्य राख्छ।

बाह्य लेखापरीक्षणले संस्थाको हिस्सा नभएको स्वतन्त्र लेखा परीक्षकद्वारा सूचना प्रणालीको मूल्याङ्कन समावेश गर्दछ। बाह्य लेखापरीक्षणको प्राथमिक उद्देश्य सूचना प्रणालीको प्रभावकारिता र दक्षताका बारेमा निष्पक्ष राय प्रदान गर्नु हो। बाह्य लेखा परीक्षकहरूले सामान्यतया कानून र नियमहरूको पालना सुनिश्चित गर्ने, जोखिम र कमजोरीहरू पहिचान गर्ने, र सुधारका लागि सिफारिसहरू सुझाव दिन लेखापरीक्षणहरू गर्छन्। बाह्य लेखापरीक्षण प्रतिवेदन सामान्यतया व्यवस्थापन र नियामक निकायहरू, लगानीकर्ताहरू र ग्राहकहरू लगायत अन्य सरोकारवालाहरूलाई पेश गरिन्छ।

सूचना प्रणालीको गुणस्तर र अखण्डता सुनिश्चित गर्न आन्तरिक र बाह्य दुवै लेखापरीक्षण आवश्यक हुन्छ। लेखापरीक्षण प्रक्रियामा सामान्यतया कागजातहरूको समीक्षा, कर्मचारीहरूको अन्तर्वार्ता, र परीक्षण प्रणाली नियन्त्रण र प्रक्रियाहरू समावेश हुन्छन्। लेखापरीक्षण निष्कर्ष र सिफारिसहरू त्यसपछि व्यवस्थापनलाई रिपोर्ट गरिन्छ, र लेखापरीक्षणको क्रममा पहिचान गरिएका कुनै पनि समस्याहरूलाई सम्बोधन गर्न सुधारात्मक कार्यहरू लिइन्छ।

Module 7: Perform Project Project Work