1. Загальні положення: найменування роботи, умовне позначення,

найменування замовника та розробника, терміни початку та закінчення

робіт.

2. Призначення системи: цілі створення програмного забезпечення, основні

очікувані результати, область застосування даного продукту.

3. Об’єкти даних: характеристика об’єктів, які мають опрацьовуватися за

допомогою розробленого програмного продукту.

4. Вимоги до програмного забезпечення: функціональні та нефункціональні

вимоги до програми. (вибрати дві нефункціональні вимоги з нового стандарту)

5. Стадії розробки: зміст робіт зі створення ПЗ (перелік етапів з короткою

характеристикою кожного).

6. Вимоги до програмної документації: перелік документів, які підлягають

розробці.

7. Посилання: перелік літератури та корисних посилань.https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/

Requirements

1. Requirements Engineering Process
2. **Requirements Elicitation**: This is the process of gathering information about the needs and expectations of stakeholders for the software system. This step involves interviews, surveys, focus groups, and other techniques to gather information from stakeholders.
3. **Requirements Analysis:** This step involves analyzing the information gathered in the requirements elicitation step to identify the high-level goals and objectives of the software system. It also involves identifying any constraints or limitations that may affect the development of the software system.
4. **Requirements Specifications:** This step involves documenting the requirements identified in the analysis step in a clear, consistent, and unambiguous manner. This step also involves prioritizing and grouping the requirements into manageable chunks.
5. Types of Requirements
6. **Functional (Behavioral) Requirements** – These are the requirements that define the functions and features of the software system. They describe what the software should do, and how it should behave when specific user actions or inputs are provided. Functional requirements are often documented as use cases or user stories.
7. **Non-functional Requirements** –define the quality attributes of the software system, such as performance, reliability, scalability, usability, security, and compatibility. Non-functional requirements are critical for ensuring that the software meets the user’s needs and expectations.

* Interface constraints
* Performance constraints: response time, security, storage space, etc.
* Operating constraints
* Life cycle constraints: maintainability, portability, etc.
* Economic constraints

**Requirements should be:**