

## **ІНФОРМАЦІЙНИЙ МЕНЕДЖМЕНТ СКЛАДНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ**

### **Вступ**

Складність сучасних інформаційних систем може сягати надскладних масштабів. Жодний працівник, менеджер чи продукт-дизайнер, що працює в подібних системах не знає іноді й її половини.

Глобальний обсяг даних зріс експоненціально з 4,4 зеттабайт до 44 зеттабайт у період з 2013 по 2020 рік. До 2025 року IDC прогнозує, що цей обсяг буде вже 163 зеттабайт.

Подібні об'єми складності сучасних систем та баз даних вимагають створення нових методів їх організації, створення та супроводу.

### **Мета роботи**

Визначити нові системи, принципи та техніки для роботи з надскладними комп'ютерними системами. Щоб досягти мети дослідження, необхідно проаналізувати існуючі методи та визначити подальший напрямок сфери ІТ в нових обставинах.

### **Результати дослідження та їх обговорення**

Один розробник здатен ефективно працювати лише над певним, обмеженим об'ємом даних, тобто над певною частиною проекту. Таким чином, одна команда та кожний розробник в ній повинен працювати над обмеженою частиною програми. Точні показники, які б визначили складність певної системи чи алгоритму визначити складно.

Процеси менеджменту складності систем в сфері ІТ найкраще визначаються типом обраної архітектури. Найбільш популярна архітектура для створення та розвитку складних комп'ютерних систем - це багаторівнева архітектура.

В ній програма складається з ієрархічних рівнів, рівень являє собою набір модулів, що надають згуртований набір функцій, кожен рівень може використовувати лише нижні рівні, тобто рівні, від яких цей рівень залежить. Якщо використати принцип інверсії залежностей, то рівні в системі стануть більш незалежними, що збільшить автономність команд, які працюють над кожним рівнем.

Інтерфейси можуть існувати не тільки на рівні понять всередині певної мови програмування, але й як веб-протокол, тобто набір правил, згідно з яких розробники можуть спілкуватися з іншими компонентами системи. Якщо використати цей тип інтерфейсу, отримаємо другий тип архітектури для менеджменту складних систем – мікросервісну архітектуру. Ця архітектура визначає комп'ютерну систему як набір незалежних сервісів, які спілкуються один з одним через веб-інтерфейси. Основна відмінність мікросервісної архітектури від багаторівневої – це тип інтерфейсів через які компоненти системи спілкуються між собою.

Веб-інтерфейси змінити важче, через що вони краще запобігають зливанню компонентів один з одним в надто складні компоненти, які потім важко підтримувати. Таким чином, вибір цих двох архітектур повинен полягати в наступному питанні: «наскільки чітко та суворо повинні бути встановлені межі між певними компонентами системи»? Чим суворіші ці межі, тим кращий захист від того, що компоненти не стануть зливатися один з одним та ставати надто складними для підтримки, чим легші ці межі - тим легше ці компоненти можна бути реорганізувати.

### **Висновки**

Способи організації складної системи засновані на організації її компонентів та визначені інтерфейсів між ними. Тож, визначення підходящої архітектури для комп'ютерної системи та її компонентів є ключовим в процесі менеджменту складності цієї системи в майбутньому.

### **Список використаних джерел**

1. <https://cacm.acm.org/magazines/2016/7/204032-why-google-stores-billions-of-lines-of-code-in-a-single-repository/fulltext>.
2. <https://softwareengineering.stackexchange.com/questions/298117/why-are-the-sizes-of-programs-so-large>.
3. <https://www.quastor.org/p/how-whatsapp-scaled-to-1-billion>.