

Розробка та використання інформаційних мереж

Викладач:

Жереб Костянтин Анатолійович

zhereb@gmail.com

Інформація про курс

- 1 семестр
- Лекції та лабораторні роботи
 - Заплановано 8-9 лекцій, 8-10 лабораторних
- Завдання по лабораторним (**40 балів**)
 - Інший викладач
- Навчальний проект (**20 балів**)
- Бонусні бали
- Іспит (**40 балів**)
- Автомат?
 - ~ **55 балів** в семестрі (лабораторні + проект + бонус)
 - **Здати все вчасно!**

Дати (лекції)

вересень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

жовтень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

листопад 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

грудень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

- Заплановано 9 лекцій
- Але залишається лише 8 “непарних” тижнів
- Аналогічно, заплановано 10 лабораторних
- Але лише 9 “парних” тижнів
- Також 16 жовтня начебто вихідний? (перенесено з 14 жовтня)

Особливості курсу: лекції

- Слайди лекцій надсилаються на email
 - Інколи додаткові матеріали
- Лекції оглядового характеру
- Рекомендації щодо правильного використання технологій
- Очікується активна участь в обговоренні
 - За це будуть бонусні бали





Навчальний проект

Попередній досвід навчальних проектів?

Вибір теми?

пропонували свою тему, викладач схвалює
оцінка часу - проблема

Розмір та структура команд?

3-4 оптимальний, 5 непогано
team lead + all developers

Дедлайни?

1 проект = фінальний

1 проект = проміжні, кожен тиждень, спершу багато звітів, потім за 2 тижні підняти

Вимоги до результатів роботи?

готовий застосунок

вимоги

звіти

командна презентація архітектури

тест план (вся команда, кожен пише свою частину, кожен тестує свою
+ все)

продуктивність - фінальний звіт

Оцінювання?

за кожен етап окремо

звіт = 8, презентація = 10, весь проект = 10

тім лід розподіляє бали



Попередній досвід навчальних проектів?

Які аспекти були цікавими чи корисними?

взаємодія в команді
кінцевий результат (який працює)



Які аспекти були невдалими?

використання незнайомих технологій jinja
взаємодія в команді
написання звітів - багато часу, шрифт не той, слово не те
мало часу
відсутність мотивації (нецікаві технології, нецікаві теми)



Що варто було б змінити?

краще менший проект і доробити, ніж великі вимоги і не зробити
менше ходити на пари, більше працювати над проектом
не чекати два тижні щоб задати питання

Проект

- Реалізувати веб-застосунок
- Головний результат – працюючий застосунок
 - Проте додаткові матеріали (вимоги, архітектура, презентація, ...) будуть корисними
- Можна робити індивідуально
- Можна робити в команді (орієнтовно до 4)
 - Відповідно більш серйозний проект

	Можна	Бажано	Обов'язково
Вибрати тему, технології, визначитись з вимогами	хоч зараз	до 25 вересня	до 9 жовтня
Перша демонстрація, розгорнутий проект, репозиторій з кодом	до наступної лекції	до 23 жовтня	до 6 листопада
Майже готовий проект	на наступній лекції	до 20 листопада	до 4 грудня

Дати (проект)

вересень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

жовтень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

листопад 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

грудень 2017						
нд.	пн.	вт.	ср.	чт.	пт.	сб.
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						



Бажано



Обов'язково

- Проекти можна демонструвати під час лекцій
- Або віддалено (email, TeamViewer, ...)

Дедлайни

- Дедлайни варто поважати
 - Важливе вміння, незалежно від використаних технологій, предметної галузі, ...
- Тому за пропуск **обов'язкових** дедлайнів будуть зніматись бали
 - -5 балів за кожен пропущений тиждень?

Don't Miss the
DEADLINE!

Бонусні бали



- Додатково до 60 балів за семестр
 - Можна отримати повний бал і без них
 - Але якщо десь втрачено бали – за допомогою бонусних балів можна набрати достатньо для бажаного результату
 - Або зробити “запас” балів на випадок якихось небажаних ситуацій в майбутньому
 - Або ж просто набрати дуже багато балів 😊
- Деякі способи заробити бонусні бали
 - Відповіді на додаткові питання
 - Участь в обговореннях під час лекцій
 - Вказати недоліки/помилки/незрозумілі місця в слайдах лекцій та інших матеріалах курсу
 - Інші способи?

Ви це вже знаєте?

Веб-розробка

Backend (server-side)+: MVC frameworks+? django asp.net mvc RoR spring mvc, templates, GET/POST requests+, validation+, authentication+, sessions+?

Frontend (client-side, browser): HTML+, CSS+, JavaScript?, DOM+, AJAX?, CSS tricks+?

Mobile development- Android?+ iOS-

Cloud computing-?, virtual machines+, containers+ docker?

Web services/web APIs, SOAP? and RESTful+ services

Мережеві протоколи+, модель OSI, стек TCP/IP, сокети, HTTP+ websockets+

Операційні системи?, адміністрування, Linux, command line tools

Реляційні БД+, SQL, ORM, проектування БД, індекси?, NoSql--

Безпека мереж-, криптографія+, криптографічні примітиви+ bcrypt-

Підвищення продуктивності коду (оптимізація), profilers ?

Паралельні та розподілені обчислення +

Тестування, unit tests, UI tests - +?

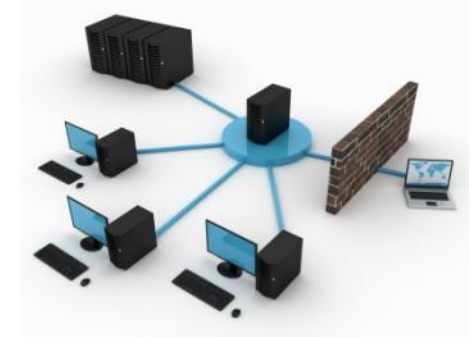
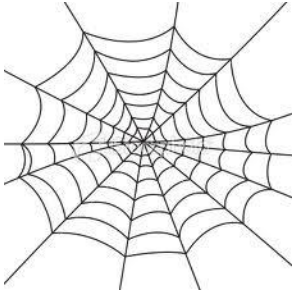
Development tools, revision control, build and deploy, continuous integration +

Software Engineering +

Processes: formal, waterfall, iterative, agile

Activities, roles and artifacts





Лекція 1.

Сучасні інформаційні мережі



1. Чому мережеві технології – це важливо?

широкий доступ
(для багатьох користувачів)
швидкий доступ
зручність
використовується в розробці
майбутнє

- Широко розповсюджені
- Попит на створення сайтів/застосунків
- Заміна інших технологій
- Нові технології, стартапи
- Зручність для користувача
- Швидкий доступ до інформації
- Швидке оновлення
- Взаємодія користувачів
- Автоматизація бізнес-процесів
- Дистанційна діяльність
- Створення контенту користувачами
- Розподілені ресурси
- ...



2. Хто цікавиться мережами?

User

- Ordinary users
- Power users
- Special needs
- International
- New/returning
- Free/paid
- Security and privacy aware



Technical

- Developer
 - Backend
 - Frontend
 - Low-level
- Designer, UI/UX
- Tester, QA
- Administrator
- Security



Business

- Product idea
- Market analysis
- Monetization, revenue model
- Promotion
- Acquisition-conversion-retention
- Analytics
- A/B testing
- Legal aspects, IP, regulations



Взаємодія ролей

- Одна людина може бути одночасно в декількох ролях
 - Розробник певного комерційного застосунку
 - Користувач засобів розробки (IDE), засобів командної комунікації, ...
 - Придумує ідею для side project
- Немає чіткого розділу між ролями
 - API: користувач чи розробник?
- Варто розуміти точку зору інших ролей
 - Це допоможе краще виконувати свою роль
- Не варто обмежуватись лише розробкою за готовою специфікацією

3. Які технології зараз важливі/популярні?

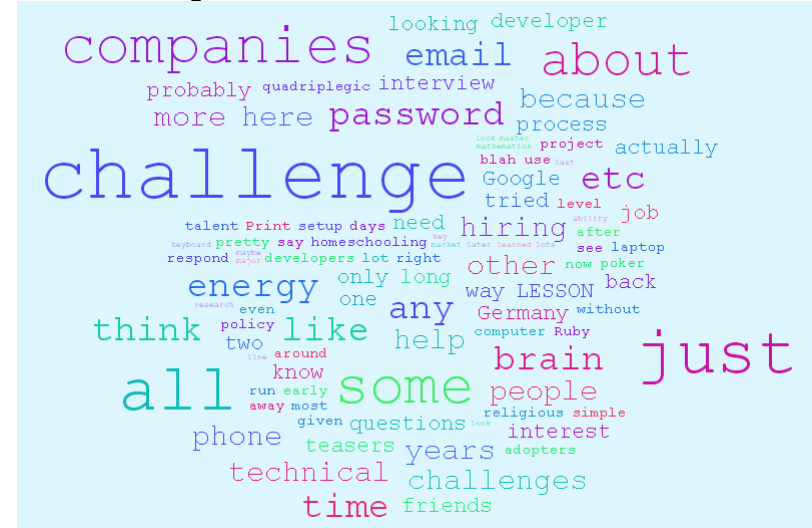
HTTP
JavaScript
CSS
Angular
React, Vue
Python, Django
C++
Java, Android

3. Які технології зараз важливі?



4. Чому не все так просто?

1. Скорочення часу розробки
2. Швидка зміна технологій
3. Складність системи
4. Різноманітність систем
5. Забезпечення надійності
6. Забезпечення продуктивності
7. Безпека



4.1. Скорочення часу розробки

- Очікування швидкої розробки
 - Тижні-місяці замість років
- Треба встигнути раніше конкурентів
 - “Ідеальний” продукт, реалізований пізніше, не зможе перехопити користувачів у менш ідеальних конкурентів, що випустили раніше
- Швидкі ітерації
 - Демонстрація проміжних варіантів – дозволяє переконатись, що продукт підходить замовнику
 - Швидке реагування на зміни вимог



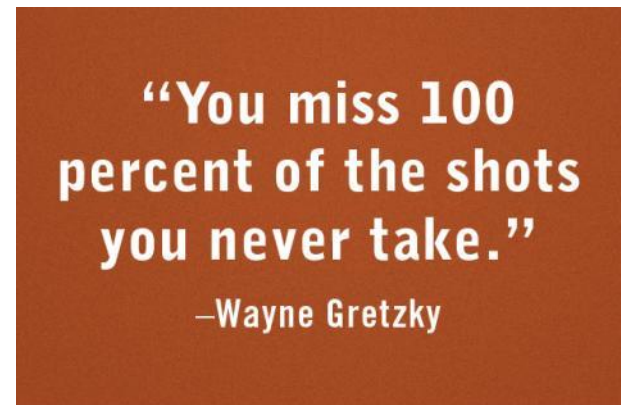
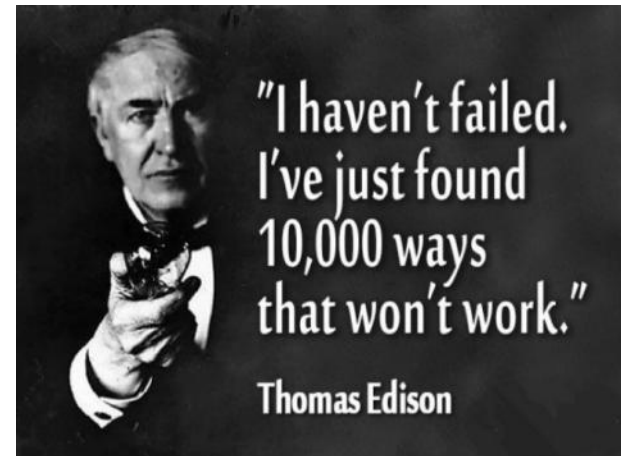
Fail Fast

Що це означає?



Fail Fast

- What it *does not* mean
 - Failure is good
 - Failure has no costs
 - Failing customers is OK
- What it *does* mean
 - Fast iterations
 - Hard and risky parts first
 - Learn from failure



Швидкість важлива

- Вірогідність зробити щось неправильно завжди буде
- Але краще помітити це раніше – і виправити
- Якщо витратити багато часу
 - Вірогідність уникнути помилки зростає несуттєво
 - Але не залишається можливості виправити помилку



З чого починати розробку проекту

GETTING
STARTED
IS THE
HARDEST
PART.

- Антипатерн
 - Починаємо розробку з того, що просто і зрозуміло
 - Створюємо веб сайт, БД, мобільний клієнт, RESTful API, SEO, ...
 - Ключовий алгоритм залишаємо на останнє
- Найбільш складні частини мають більше шансів викликати проблеми
 - Краще про це дізнатись раніше
- Більша невизначеність з часом реалізації
- Якщо почати думати над складними частинами раніше – вони довше “просидять у підсвідомості”



Не вчитись на помилках – значить їх повторювати

- З невдалих проектів – хоч якась користь
- Знання, що так не працює – теж важливе знання
- Ретроспектива / post-mortem

*it's only a mistake
if you don't learn from it*

Особливості курсу:

навчальні проекти

- Основна мета – навчитись, а не здати абищо
 - Не варто обирати занадто прості проекти
 - Можна обирати технології, з якими ще не працювали
 - Краще цікавий проекти, в якому не все вийде – ніж нудний проект, який вже здавався безліч разів
- Краще фокусуватись не на кількості, а на якості
 - Варіантів використання, веб-сторінок, таблиць в БД, ...
- На лекціях не буде детальних пояснень, як реалізовувати проект
 - Варто читати документацію та інші джерела
 - Але якщо виникають питання – треба одразу їх задавати!



Knowledge is
having the right
answer.

Intelligence is
asking the right
question.

4.2. Швидка зміна технологій

- Постійно з'являються нові фреймворки
 - Особливо у веб-розробці
- Часто нові підходи, принципи роботи, архітектури
- Часу на вивчення дуже мало
- Треба вчитись швидко вчитись



Особливості курсу: лекції

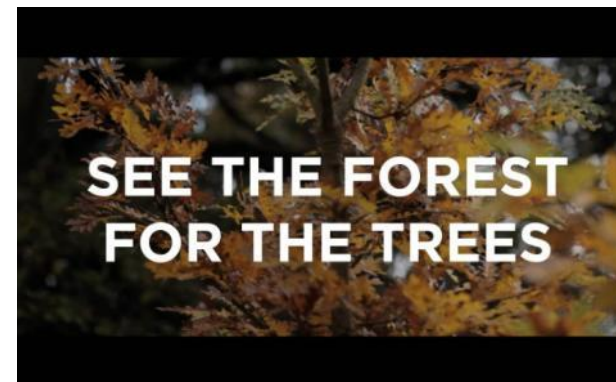
- Слайди лекцій надсилаються на email
 - Інколи додаткові матеріали
- **Лекції оглядового характеру**
- Рекомендації щодо правильного використання технологій
- Очікується активна участь в обговоренні
 - За це будуть бонусні бали

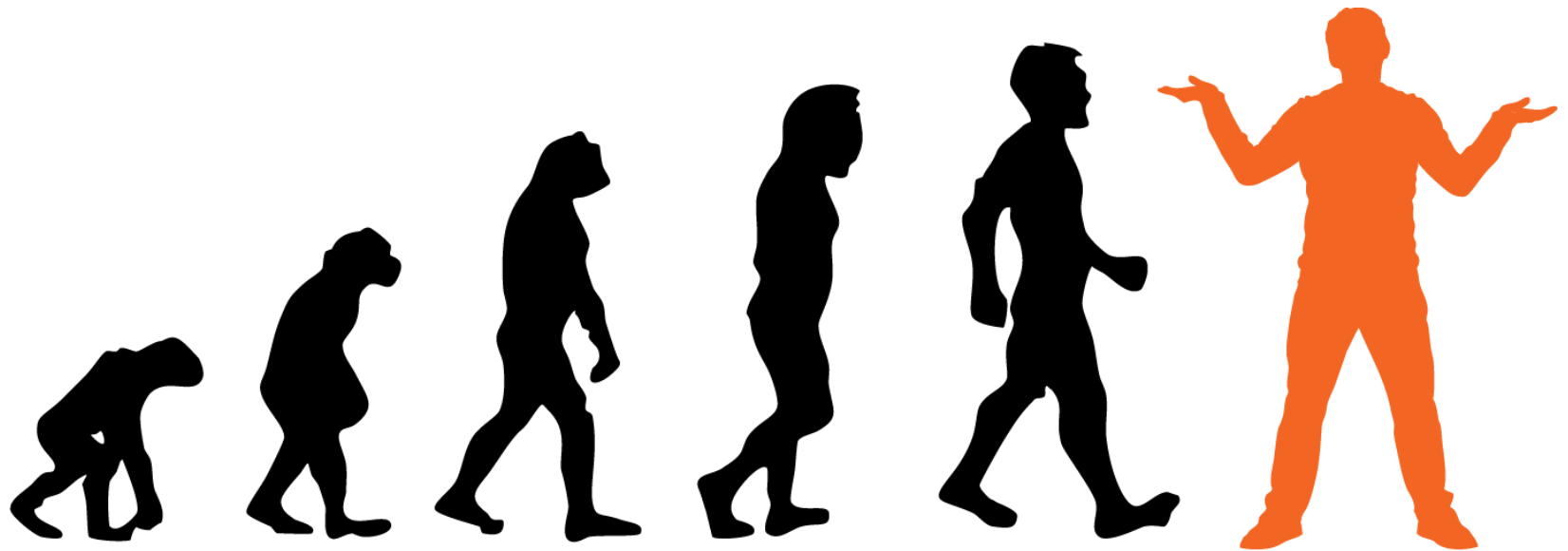


Чому оглядові лекції

(персональна точка зору)

- Безліч технологій
- Швидкі зміни
- Для більшості технологій доступна якісна документація
 - А також StackOverflow, форуми/блоги, Wikipedia, Google, ...
 - Корисно розвивати вміння їх читати 😊
- Більш загальні і корисні знання
 - Базові принципи
 - Які напрями взагалі існують
 - Best practices





Learning to Learn

4.3. Складність

- Багато компонентів
- Асинхронна взаємодія
- Важко відлагоджувати
- Швидка зміна технологій
- Потрібен комплексний підхід до побудови мережі



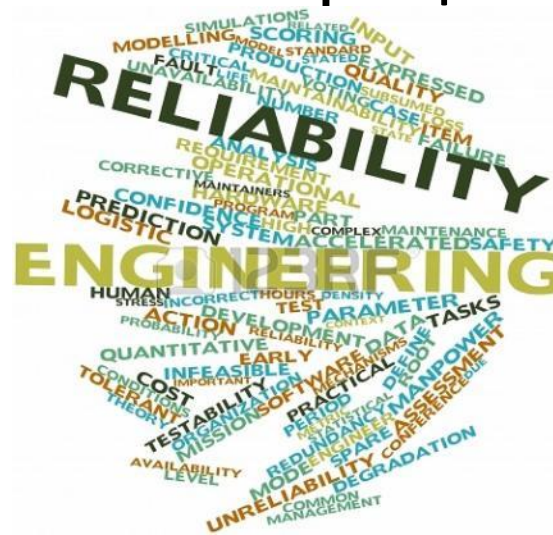
4.4. Різноманітність

- Апаратні платформи
- Операційні системи
- Формати даних
- Протоколи
- Конфігурації мережі
- Версії компонент
- Браузери
- Мобільні клієнти



4.5. Надійність

- Фізичні мережі можуть бути ненадійними
- Протоколи нижчих рівнів можуть втрачати пакети
- Деякі вузли мережі можуть виходити з ладу
- Проте система має працювати стабільно

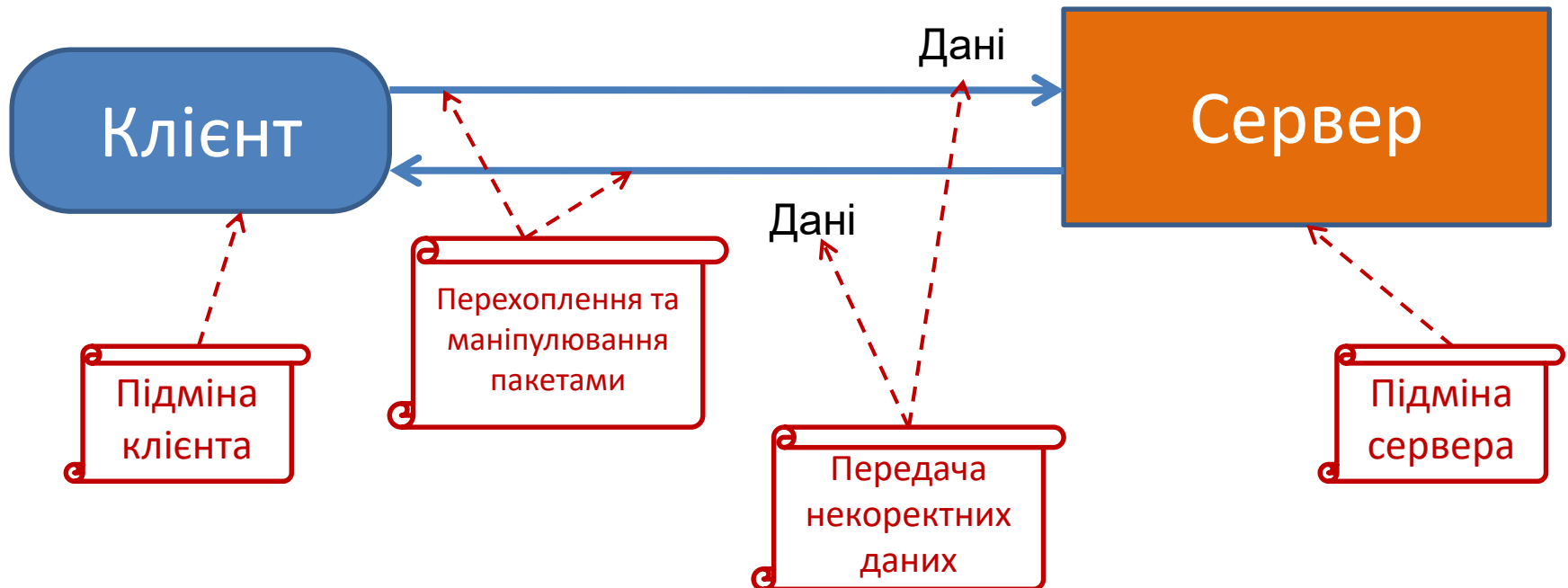


4.6. Продуктивність

- Затримки передачі даних
- Повторне посилання втрачених пакетів
- Timeouts
- Розподіл обчислень між клієнтом та сервером
- Велика кількість запитів, масштабування
- Коливання рівня завантаженості
- Рівні якості послуг (QoS)



4.7. Безпека



- Базові протоколи розроблені без урахування безпеки

Подальші кроки



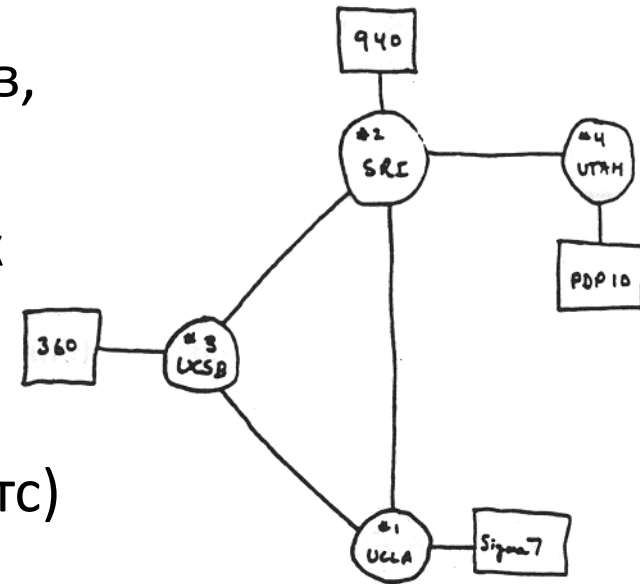
- Думати над темою проекту
 - Дедлайн вже близько...
- “Brainstorming” тем проектів **(+бонусні бали)**
 - Будуть надіслані певні ідеї проектів
 - Завдання: придумати нові теми та/або прокоментувати вже надіслані
 - Завдання не є обов’язковим, проте дає можливість отримати бонусні бали
- Ретроспектива попередніх проектів **(+бонусні бали)**
 - Підготувати ретроспективу раніше виконаних (чи не виконаних) проектів
 - Можна робити самостійно чи командою
 - Можна зробити для декількох проектів
 - Завдання не є обов’язковим, проте дає можливість отримати бонусні бали
- Опитування **(+бонусні бали)**
 - Буде надіслано форму для опитування
 - Мета: врахувати побажання студентів щодо курсу
 - Велике прохання вчасно прийняти участь в опитуванні – це допоможе зробити курс більш цікавим та корисним

Висновки

- Інформаційні мережі застосовуються у всіх сферах життя
- Бурхливий розвиток
- Різні ролі відносно мереж – не треба обмежуватись лише однією
- Виклики при розробці та впровадженні
– ... і як з ними боротись (в наступних лекціях...)

Історія мереж-1

- 1960-і роки – принцип комутації пакетів, перші мережі розподілу часу
- 1962 – ідея глобальної мережі, початок роботи над мережами в ARPA
- 1965 – зв'язок між комп'ютерами на далекі відстані (Каліфорнія – Масачусетс)
- 1969 – мережа ARPANET



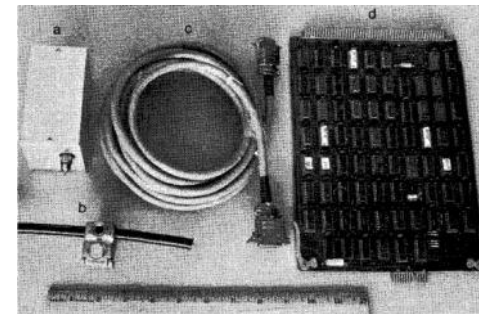
THE ARPA NETWORK

DEC 1969

4 NODES

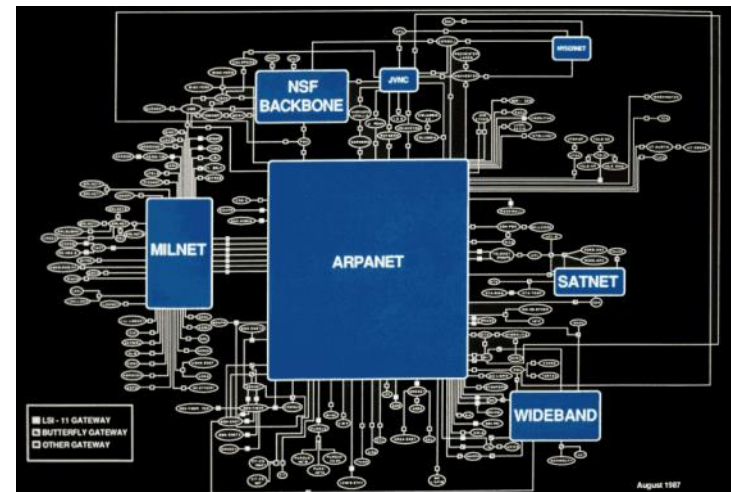
Історія мереж-2

- 1970 – протокол Network Control Program
- 1971 – електронна пошта, символ @
- 1971 – протокол FTP
- 1974 – протокол TCP
- 1974 – перша згадка терміну Internet
- 1976 – Ethernet
- 1978 – TCP розділяється на TCP та IP
- 1983 – ARPANET переходить на TCP/IP




Історія мереж-3

- 1970-80і роки – інші мережі
- 1985 – мережа NSFNET – початок сучасного Інтернету
- 1990 – технологія WorldWideWeb (мова HTML, протокол HTTP, перший веб-сервер і веб-браузер)



Історія мереж-4

- 1993 – перші засоби веб пошуку (W3Catalog, Aliweb, JumpStation)
- 1995 – перший сучасний instant messenger PowWow
- 1998 – Google web search 
- середина 1990-х – комерційне використання WWW
- ~1997-2000 – dot-com boom (bubble)

Мережі в 21 столітті

- Веб-застосунки
- Блоги
- Вікі
- Соціальні мережі, Twitter
- Messengers, chat bots

- Мобільні пристрої
- Війни браузерів
- Різноманіття засобів розробки
- Комбінування сайтів, mashups, Web APIs
- Віртуалізація, хмарні технології
- Безпека, privacy

- Peer-to-peer (P2P) мережі
- Мультимедійні застосунки (аудіо, відео)
- IP-телефонія, VoIP
- Wireless
- Онлайн ігри, VR/AR
- Smart houses, smart cars, ...
- Internet of Things
- Tor, I2P
- Cryptocurrency, Bitcoin, blockchain ...

Нові технології (buzzwords)

- Web 2.0
 - Соціальні аспекти: користувачі створюють контент
 - Технологічні аспекти: AJAX, DHTML, XML/JSON
- Rich Internet Applications (RIA)
 - Flash, Silverlight
- HTML5
- Content Management Systems (CMS)
- Service Oriented Architecture (SOA)
- Cloud computing
 - SaaS (Software as a Service), IaaS, PaaS, ...
- Мобільні клієнти
- Wearable
- Internet of Things



Спеціалізовані мережі

- Корпоративні мережі
- Обчислювальні мережі, кластери, Грід
- Semantic web
- Мультиагентні системи
- Сенсорні мережі



