

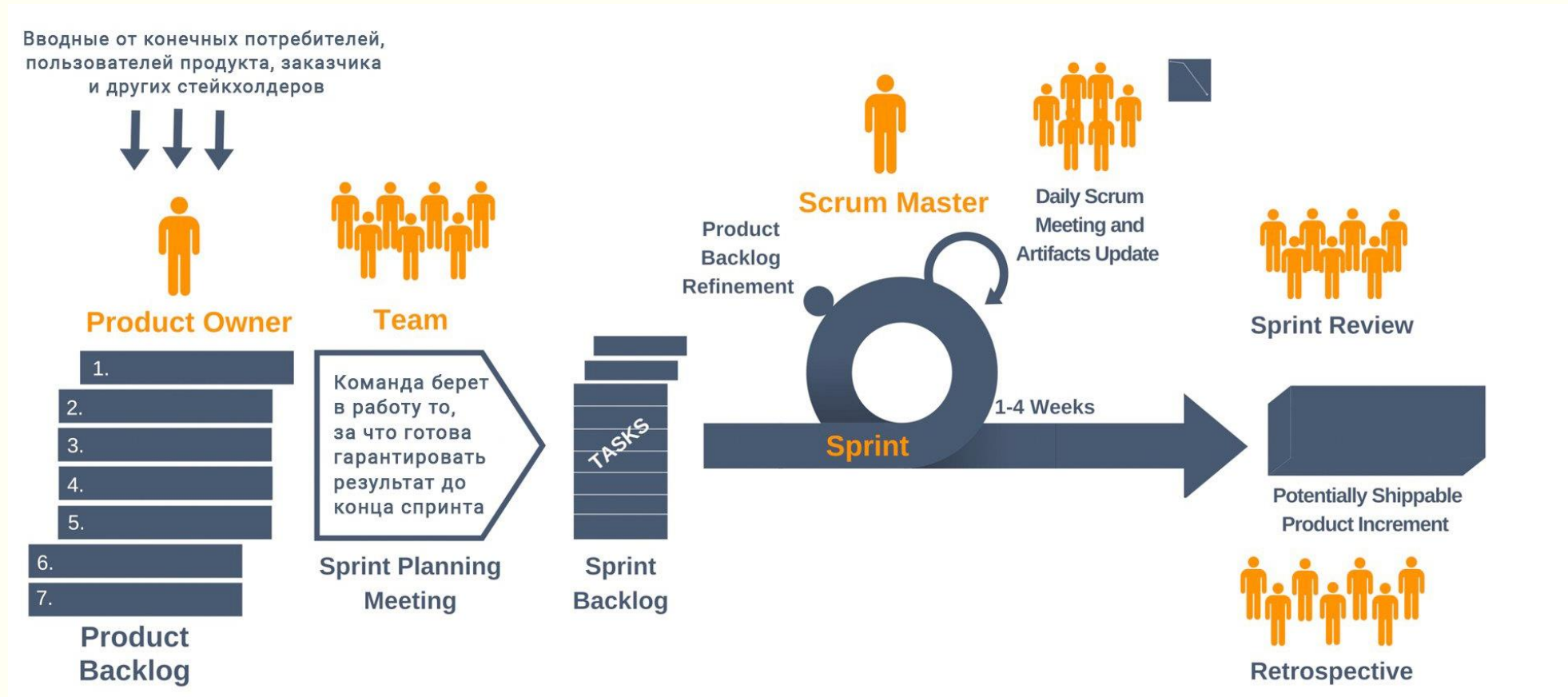


# ЕКОСИСТЕМА РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

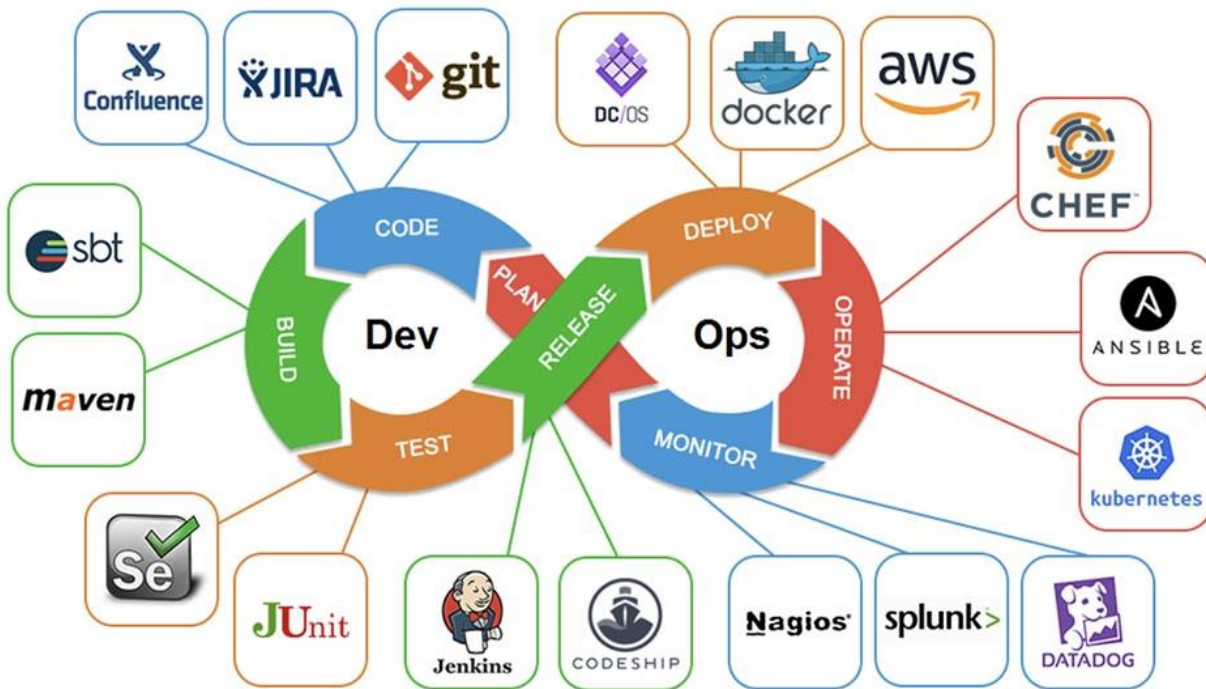
Питання 1.3.

# Scrum-підхід до організації процесу розробки ПЗ

- Сучасні Agile-підходи: Scrum, DevOps та ін.

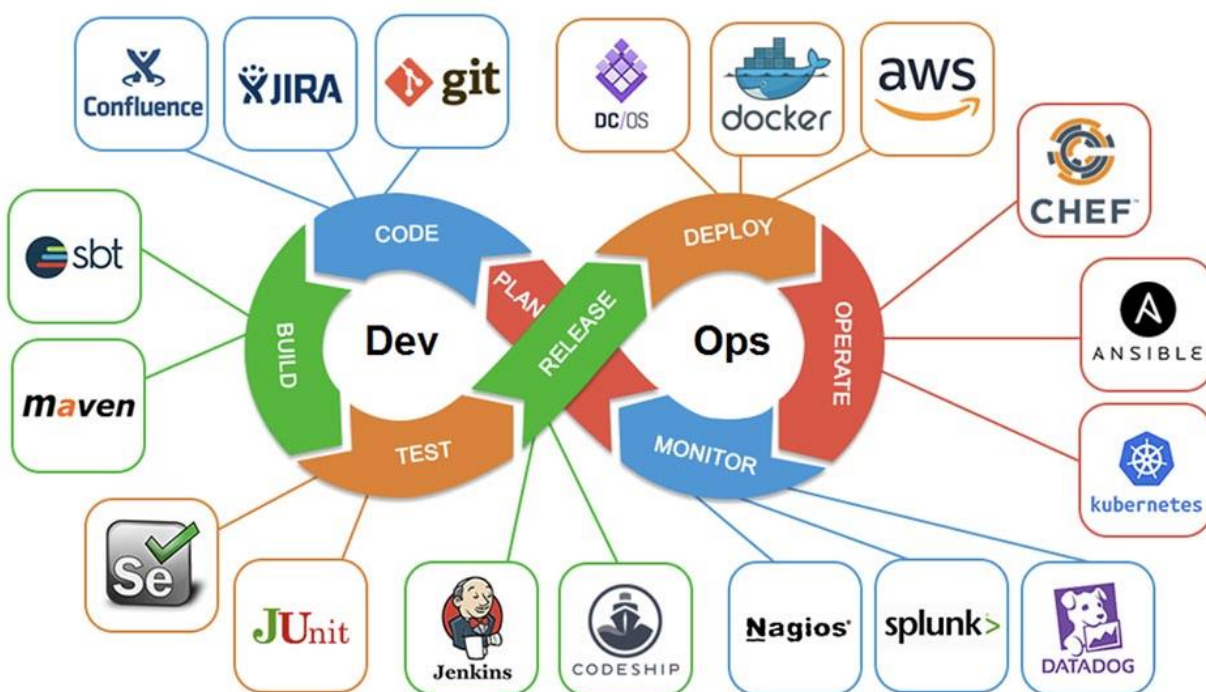


# DevOps-підхід до організації процесу розробки ПЗ



- Умовно, в DevOps-культурі можна виділити кілька ролей, які добре співвідносяться з професіями:
  - **Build Engineer** — людина, яка відповідає за збирання коду: підтягування залежностей, розбір конфліктів у коді тощо.
  - **Release Engineer** — відповідає за доставку коду з розробки в продакшн: яка вітка піде в тестування, яка збірка (білд) потрапить у продакшн та ін.
  - **Automation Engineer** — інженер з автоматизації: автоматичне збирання при пуші в Git, прогон тестів, розгортання на staging-оточенні, деплой в продакшн. Ключова роль в DevOps-підході.

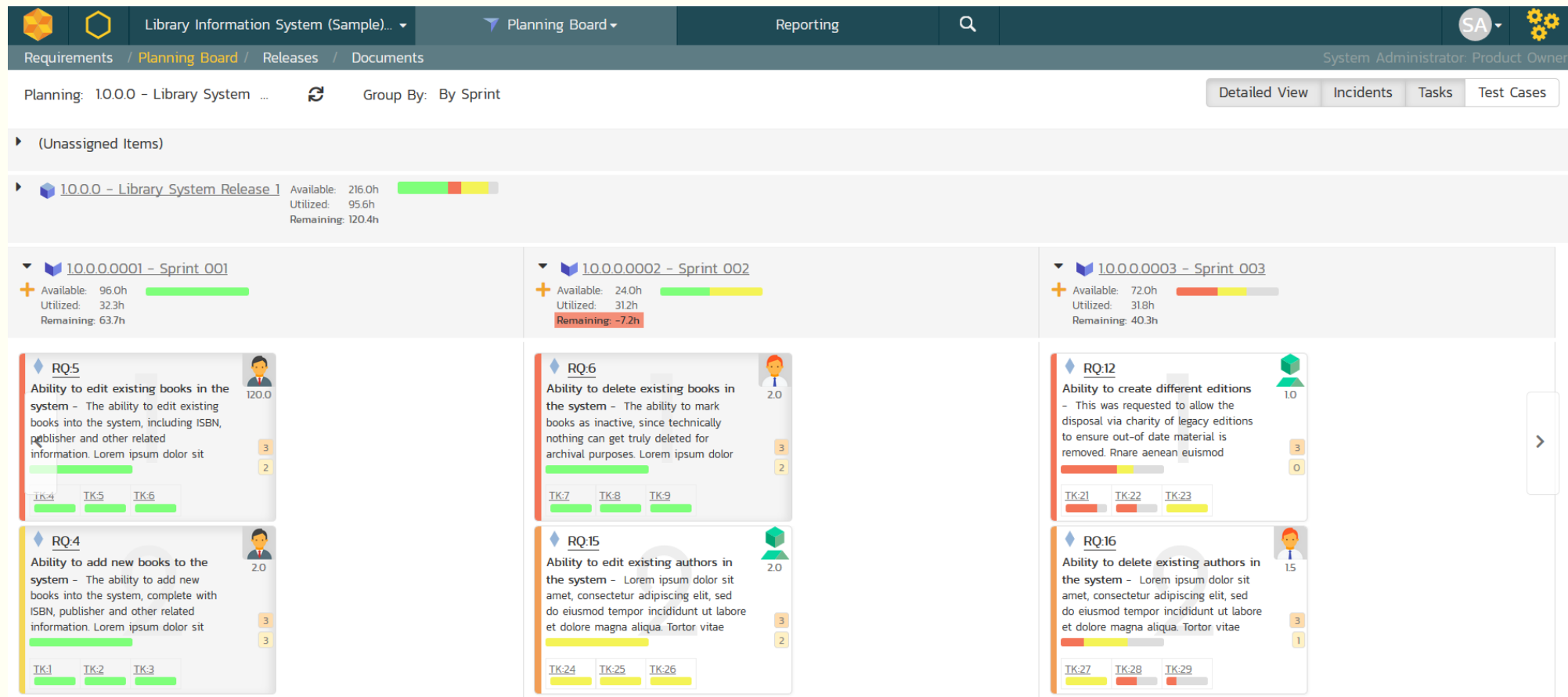
# Екосистема розробки програмного забезпечення



- У цілому, можна виділити кілька обов'язкових етапів розробки ПЗ:
  - **Планування:** аналіз вимог, постановка та трекінг задач, колаборація.
  - **Розробка:** системи контролю версій, керування репозиторіями, неперервна інтеграція, інструменти збирання коду.
  - **Тестування:** функціональне тестування, управління даними, тестування продуктивності, тестування у віртуальних середовищах (service virtualization) та ін.
  - **Випуск (реліз):** розгортання, керування конфігураціями, контейнеризація, робота з хмарою, публікація релізу.
  - **Операційне обслуговування:** моніторинг, дослідження користувацького досвіду, бізнес-аналітика.

# Планування. Аналіз вимог

- Аналіз технічних бізнес-вимог гарантує, що задачі будуть поставлені правильно, і замовник на практиці отримає те, що йому потрібно.
  - Приклад: користувацькі історії (user stories) в Infectra SpiraTeam – Kanban-дошка.





# Для не-Agile-підходів зазвичай створюється документ зі специфікаціями вимог

The screenshot shows the SpiraPlan web application interface. The top navigation bar includes 'Requirements', 'Planning Board', 'Releases', and 'Documents'. The left sidebar shows a 'Requirements Selector' with a tree view of requirements. The main content area is titled 'Requirements Document' and displays the details for 'Functional System Requirements [RQ:1]'. The status is 'In Progress', type is 'Epic', and importance is '(None)'. The text describes a System Requirements Specification (SRS) and lists child requirements like 'Online Library Management System [RQ:2]'. A note at the bottom states: 'This is not a real application, but is just a sample application used for demonstrating t'.

Requirements Document Beta

Functional System Requirements [RQ:1]

Status: In Progress Type: Epic Importance: (None)

A System Requirements Specification (SRS) (also known as a Software Requirements Specification, behavior of a system or software application. It includes a variety of elements (see below) that att satisfy their different users.

This section usually consists of a hierarchical organization of requirements, with the business/fun requirements listed as their child items. Generally the requirements are written as statements such information included as necessary.

Online Library Management System [RQ:2]

Status: In Progress Type: Epic Importance: (None)

This sample application lets you view, create and edit books in the library catalog as well as v

To view the library catalog or the authors list you will need to login as a borrower and to ma librarian.

Note: This is not a real application, but is just a sample application used for demonstrating t

- SpiraTest [test management system](#)
- SpiraTeam [application lifecycle management system](#)

The screenshot shows the SpiraPlan web application interface. The top navigation bar includes 'Requirements', 'Planning Board', 'Releases', and 'Documents'. The left sidebar shows a 'Requirements Selector' with a tree view of requirements. The main content area is titled 'Requirements Document' and displays the details for 'Creating a new book in the system [RQ:30]'. The status is 'Completed', type is 'Use Case', and importance is '(None)'. The text describes the use case and includes a flowchart with three steps: 'User logs into the system [RS:1]', 'User chooses option to create new book [RS:2]', and 'User enters books name and author [RS:3]'. The flowchart starts with a 'Start' node and ends with a loop back to the beginning.

Requirements Document Beta

Creating a new book in the system [RQ:30]

Status: Completed Type: Use Case Importance: (None)

This use case defines the steps in creating a new book.

```
graph TD; Start([Start]) --> RS1[User logs into the system [RS:1]]; RS1 --> RS2[User chooses option to create new book [RS:2]]; RS2 --> RS3[User enters books name and author [RS:3]]; RS3 --> Start;
```

# Планування. Постановка та трекінг задач

## ■ Приклад таск трекера Trello

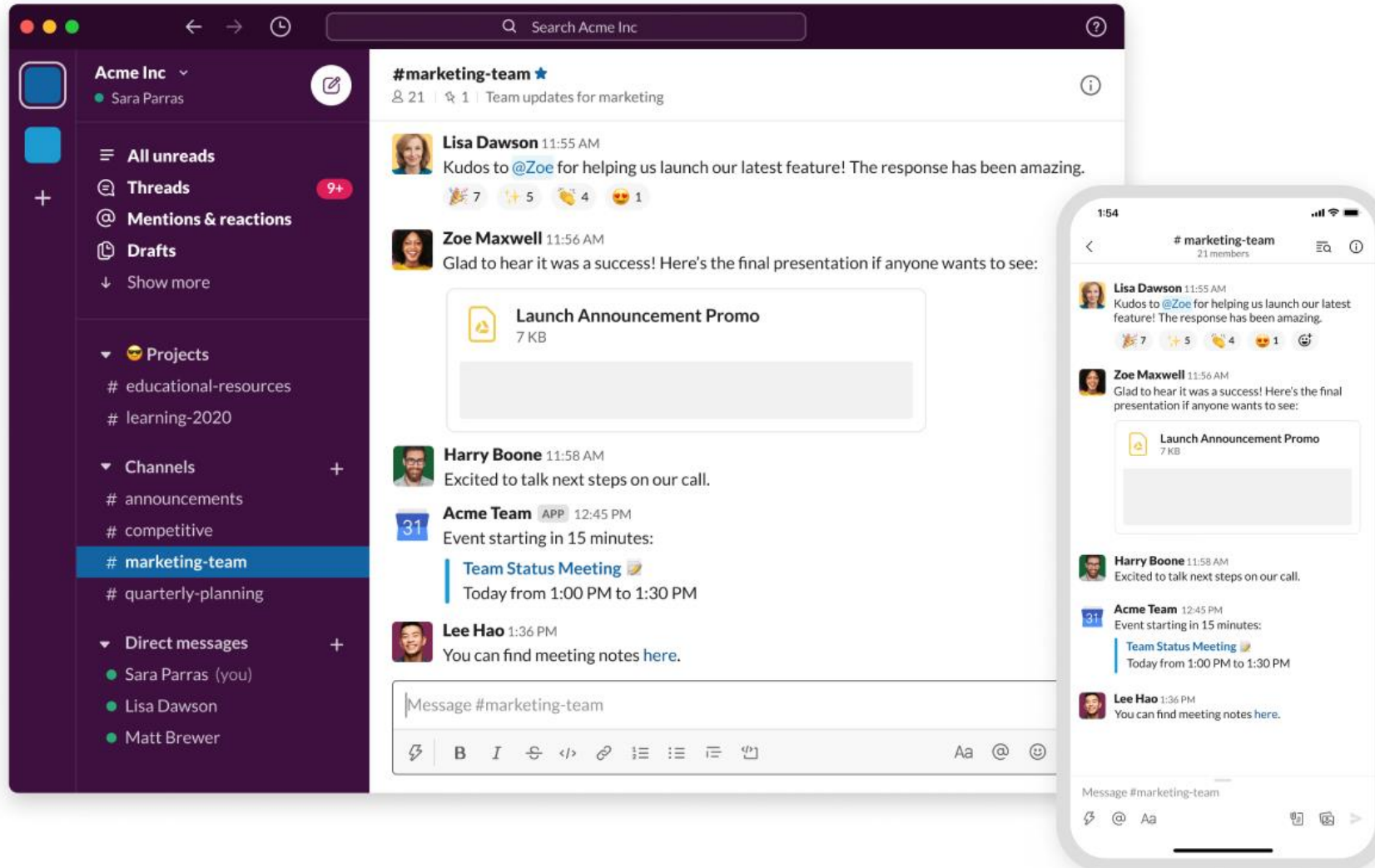
The screenshot displays the Trello Task Tracker interface. At the top, there's a navigation bar with links for 'ГЛАВНАЯ СТРАНИЦА', 'ТУР', and 'БЛОГ'. The Trello logo is centered, and on the right are buttons for 'Зарегистрироваться' and 'Войти'. Below this is a yellow banner with the text 'Хотите подписаться на эти карточки?' and a 'Бесплатная регистрация' button.

The main workspace is titled 'Task Tracker' and has tabs for 'Персональная' and 'Публичная'. A 'Меню' button is in the top right. The board is organized into four columns:

- Future:** Contains five task cards. The first card is 'Possible strategy to improve startup performance'. The second is 'Keyboard shortcut "/" puts cursor into search box'. The third is 'Performance suggestion: "Freezing" collapsed tasks'. The fourth is 'Avoid page load flash'. The fifth is 'Responsive Task List'. The bottom card is 'Global Errors Handler'.
- To Do:** Contains four task cards. The first is 'Remember list view mode when returning from Task'. The second is 'REST Strategy'. The third is 'Strategy for Required Field being blank'. The fourth is 'Quick-one: jshint unused:true'. The bottom card is 'Improve column resizing for Work Items in Task List'.
- Doing:** Contains three task cards. The first is 'Issues of Task Tracker API'. The second is 'Create and update records on backend server'. The third is '/List page improvements'.
- Awaiting Review:** Contains one task card: 'Implement list view'.

Each task card features a progress bar at the top, a title, and metadata including a list icon, comment count, checkmark status, and assignee avatars.

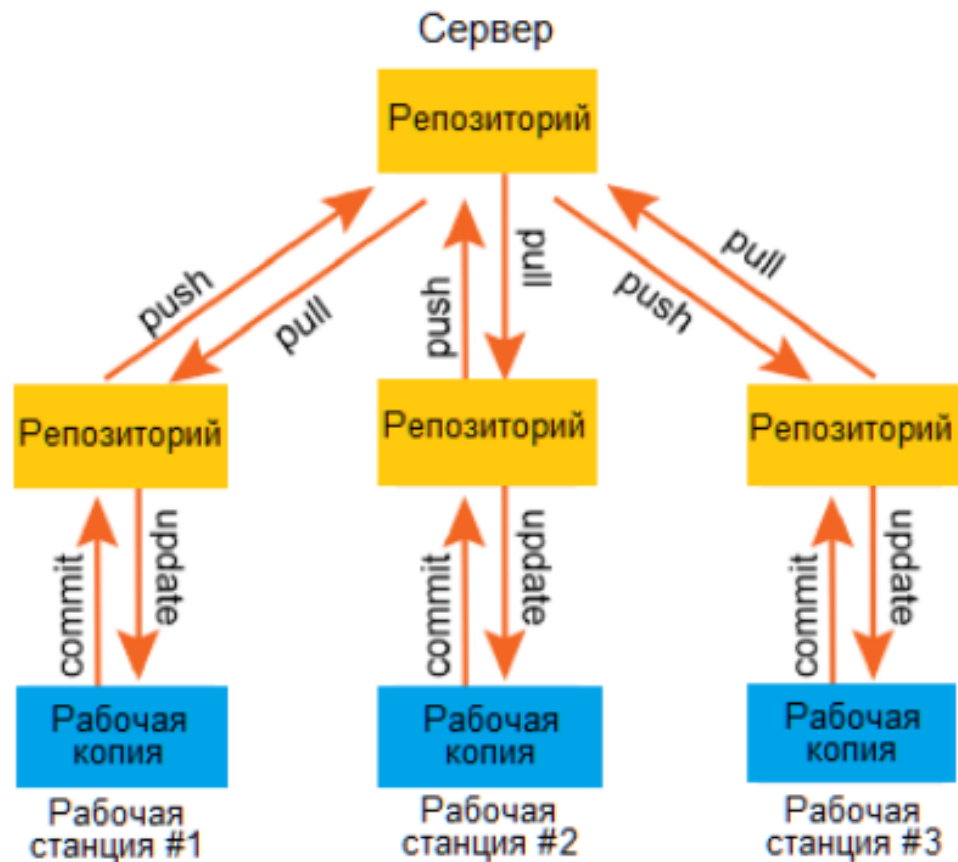
# Планування. Колаборація в проекті



- Приклад спілкування щодо проекту в Slack.



# Розробка. Системи контролю версій



## ▪ Найпопулярніша система контролю версій – Git.

- GitHub — сервіс онлайн-хостингу репозиторіїв, який володіє всіма функціями розподіленого контролю версій та функціональністю керування первинним кодом.

csbc-edu / csbc-edu.github.io

<> Code ⓘ Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

🔗 master 🔗 1 branch 🔗 0 tags

Go to file Add file ▼ Code ▼

csbc-edu Тема 1 готова ✓ 391b62d 20 hours ago 🕒 212 commits

images/menu	new frontend (thanks to sallexx)	10 months ago
java	++	4 months ago
mobile-dev	(andro) update lab 06	4 months ago
oop	Тема 1 готова	20 hours ago
os	++upd	4 months ago
programming-essentials	Змінено предмет викладання на ООП	yesterday
styles	landing update	4 months ago
visual-programming-tools	(ІЗВП) Додано лаб 06	9 months ago
index.html	lp update	yesterday

# Розробка. Неперервна інтеграція та збирання проекту

- Демо-проект Travis CI + GitHub
  - Приклад збирання проекту за допомогою Travis CI

juampynr / drupal8-travis-ci

build passing

Current

Branches

Build History

Pull Requests

> Build #7

More options

✓ Pull Request #1 Sample pull request to show Travis CI integration

Commit 321ccaf

#1: Sample pull request to show Travis CI integration

Branch master

Juampy NR authored

GitHub committed

#7 passed

Ran for 4 min 8 sec

Total time 7 min 48 sec

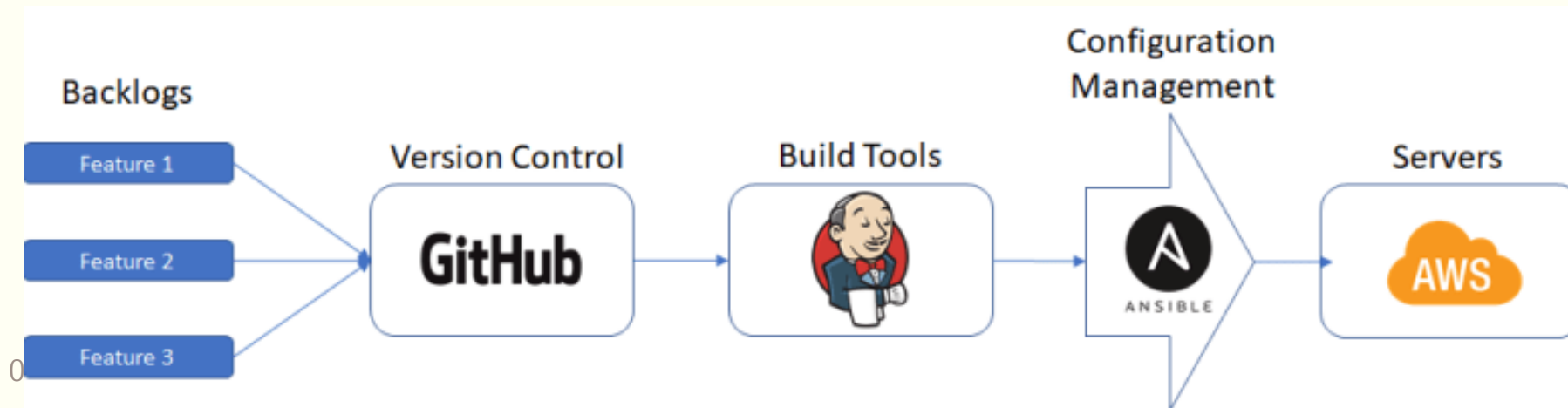
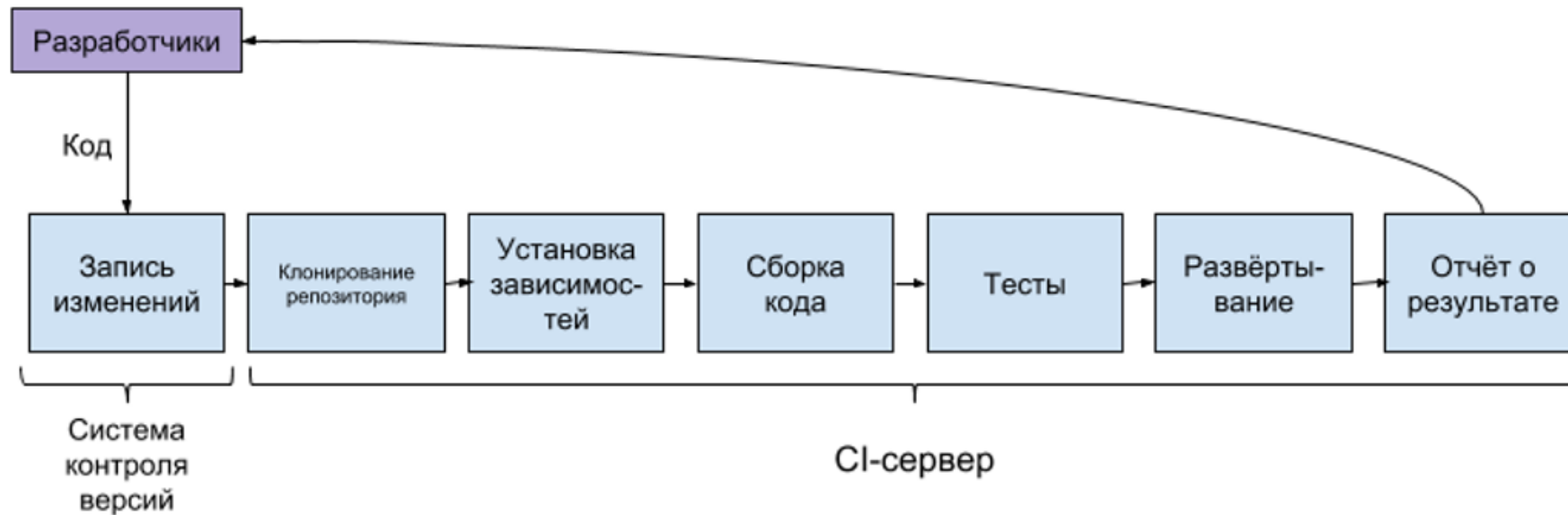
less than a minute ago

Restart build

Build Jobs

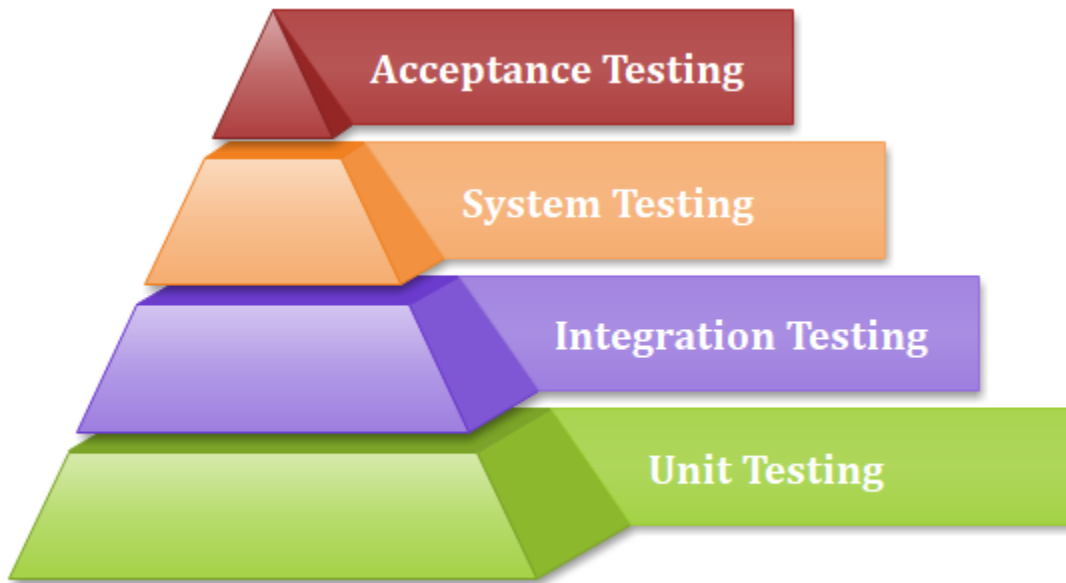
✓ # 7.1	PHP: 7.1	JOB=job:check-coding-standards	1 min 39 sec	
✓ # 7.2	PHP: 7.1	JOB=job:run-unit-tests	2 min 40 sec	
✓ # 7.3	PHP: 7.1	JOB=job:run-behat-tests	3 min 29 sec	

# Розробка. Схематика неперервної інтеграції



# Тестування. Ієрархія тестів

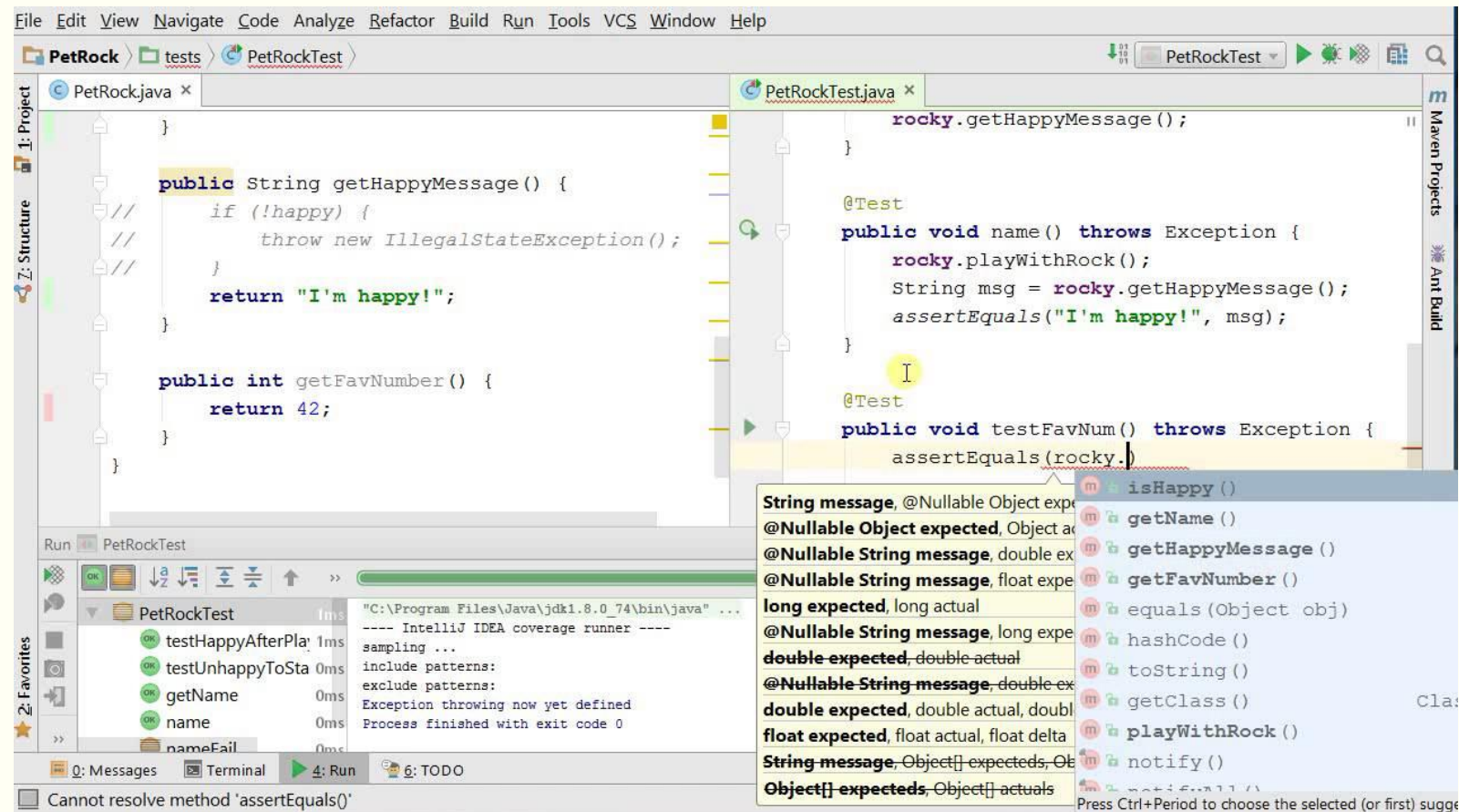
---



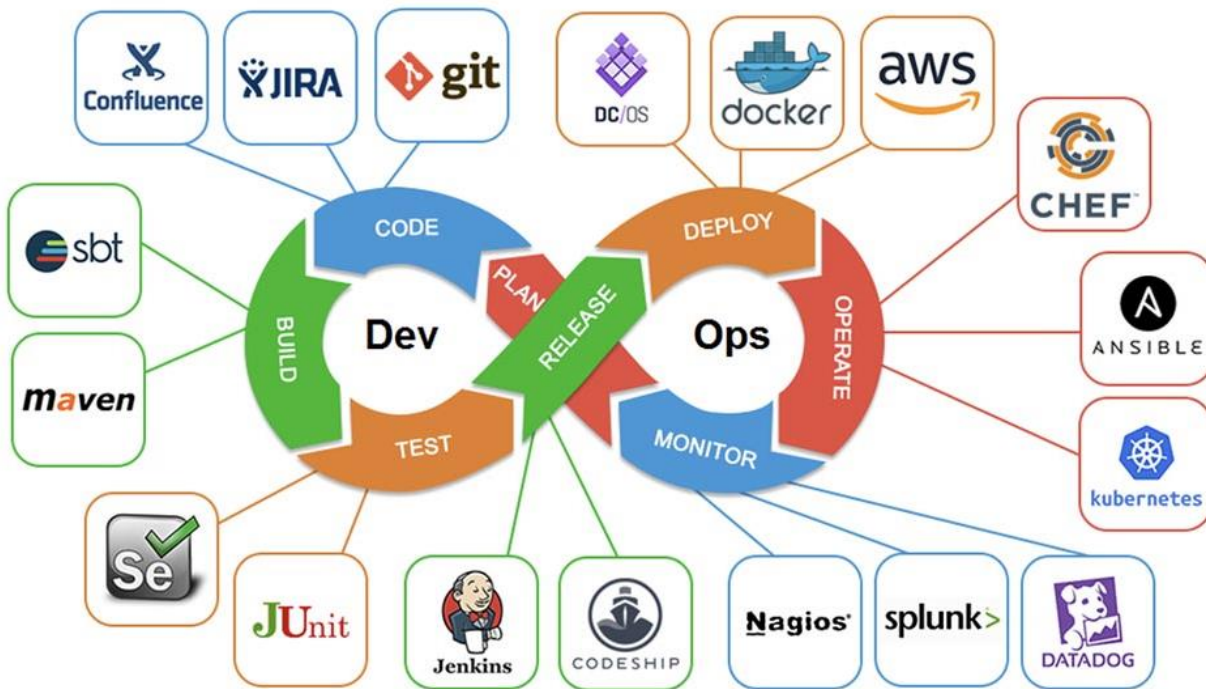
- **Модульне (*unit*) тестування** відбувається на ізольованих частинах (модулях) коду, незалежних від бібліотек та фреймворків.
  - Перевіряє бізнес-логіку додатку, допомагає документувати код, відловлює баги на ранніх стадіях розробки.
  - Займає близько 70% від усіх тестів.
- **Інтеграційне тестування** перевіряє, як окремі програмні компоненти інтегруються в програмні підсистеми.
  - Часто на цьому рівні автоматизовано тестується графічний інтерфейс користувача.
  - Займає до 20% від усіх тестів.
- **Системне тестування** забезпечує коректність роботи всієї програмної системи.
  - Зазвичай тестується зручність використання додатку.
- **Приймальне (*acceptance*) тестування** перевіряє відповідність розробленої системи вимогам замовника.
  - Етапи альфа-, бета-тестування.

# Тестування. Модульні тести

- Є набором створених програмістом в окремому проєкті функцій, які перевіряють співпадіння між очікуваними результатами роботи блоків коду та реальними.
- Приклад: мова програмування Java і тестовий фреймворк JUnit



# Випуск. Управління релізами (Release management)



- Накладається на етап планування. Передбачає
  - планування можливостей (фіч), які включатимуться в кожну релізну версію ПЗ,
  - організацію неперервного постачання (continuous delivery) нових збірок,
  - розробку плану тестування для забезпечення стабільності релізу,
  - моніторинг дефектів і результатів тестування з метою забезпечення відповідності календарному плану релізів.



# Випуск. Управління релізами в Infectra SpiraTest

## ■ Планування релізів:

Library Information System (Sample)...

Releases

Reporting

SA

Requirements / Releases / Documents

System Administrator: Product Owner

+ Insert

▼

🗑 Delete

→ Indent

← Outdent

↻ Refresh

--- Show Level ---

▼

🔍 Filter

▼

✎ Edit

▼

⚙ Tools

▼

-- Show/hide columns --

▼

Displaying 8 out of 19 release(s) for this product.

✓	Name	Version #	Test Coverage	Start Date	End Date	Plan Effort	Task Effort	Type	Status	ID	Edit
<input type="checkbox"/>	<input type="text" value="Filter"/>	<input type="text"/>	<div>-- Any --</div>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<div>-- Any --</div>	<div>-- Any --</div>	RL	<input type="button" value="Edit"/>
<input type="checkbox"/>	Library System Release 1	10.0.0	<div></div>	6-Mar-2019	5-May-2019	216.0h	94.3h	Major Release	Completed	RL:1	<input type="button" value="Edit"/>
<input type="checkbox"/>	Library System Release 1.1	11.0.0	<div></div>	6-May-2019	22-Jul-2019	168.0h	86.9h	Major Release	In Progress	RL:4	<input type="button" value="Edit"/>
<input type="checkbox"/>	Library System Release 1.1 SP1	11.1.0	<div></div>	23-May-2019	21-Jun-2019	336.0h	0.9h	Minor Release	In Progress	RL:5	<input type="button" value="Edit"/>
<input type="checkbox"/>	Library System Release 1.1 SP2	11.2.0	<div></div>	22-Jun-2019	22-Jul-2019	320.0h		Minor Release	Cancelled	RL:7	<input type="button" value="Edit"/>
<input type="checkbox"/>	Sprint 001	11.0.0.0001	<div></div>	6-May-2019	9-May-2019	24.0h	29.0h	Sprint	In Progress	RL:17	<input type="button" value="Edit"/>
<input type="checkbox"/>	Sprint 002	11.0.0.0002	<div></div>	10-May-2019	14-May-2019	72.0h	27.0h	Sprint	In Progress	RL:18	<input type="button" value="Edit"/>
<input type="checkbox"/>	Sprint 003	11.0.0.0003	<div></div>	15-May-2019	18-May-2019	72.0h	30.0h	Sprint	In Progress	RL:19	<input type="button" value="Edit"/>
<input type="checkbox"/>	Library System Release 1.2	12.0.0	<div></div>	25-Jul-2019	26-Jul-2019	16.0h	10.0h	Major Release	Planned	RL:6	<input type="button" value="Edit"/>

Show

15

▼

rows per page

⏪ ⏩ Displaying page

1

⏪ of 1 ⏩

# Випуск. Управління релізами в Infectra SpiraTest

## ■ Відстеження дефектів:

Library System Release 1.1

1.1.0.0 [RL:4] Type: Major Release Status: In Progress Operations

Overview Incidents \* Reqs & Tasks \* Test Cases \* Test Runs \* Attachments History

Display List of Incidents: Detected in This Release Refresh Filter -- Show/hide columns --

Displaying 1 - 3 out of 3 incident(s) in this release/sprint. Filtering results by Detected Release, Status. Clear Filters

✓	Name	Type	Status	Priority	Detected By	Creation Date	Owner	ID	Edit
<input type="checkbox"/>	Filter	-- Any --	Assigned	-- Any --	-- Any --		-- Any --	IN	Edit
<input type="checkbox"/>	Editing the date on a book is clunky	Bug	Assigned	2 - High	Joe P Smith	17-May-2019	Fred Bloggs	IN:8	Edit
<input type="checkbox"/>	Ability to be accessed by Mozilla	Issue	Assigned	2 - High	Joe P Smith	16-May-2019	System Administrator	IN:32	Edit
<input type="checkbox"/>	Editing the date on an author is clunky	Bug	Assigned	3 - Medium	Joe P Smith	26-May-2019	Joe P Smith	IN:9	Edit

Show 15 rows per page

Displaying page 1 of 1

## ■ та результатів тестування

Library System Release 1.1

1.1.0.0 [RL:4] Type: Major Release Status: In Progress Operations

Overview Incidents \* Reqs & Tasks \* Test Cases \* Test Runs \* Attachments History

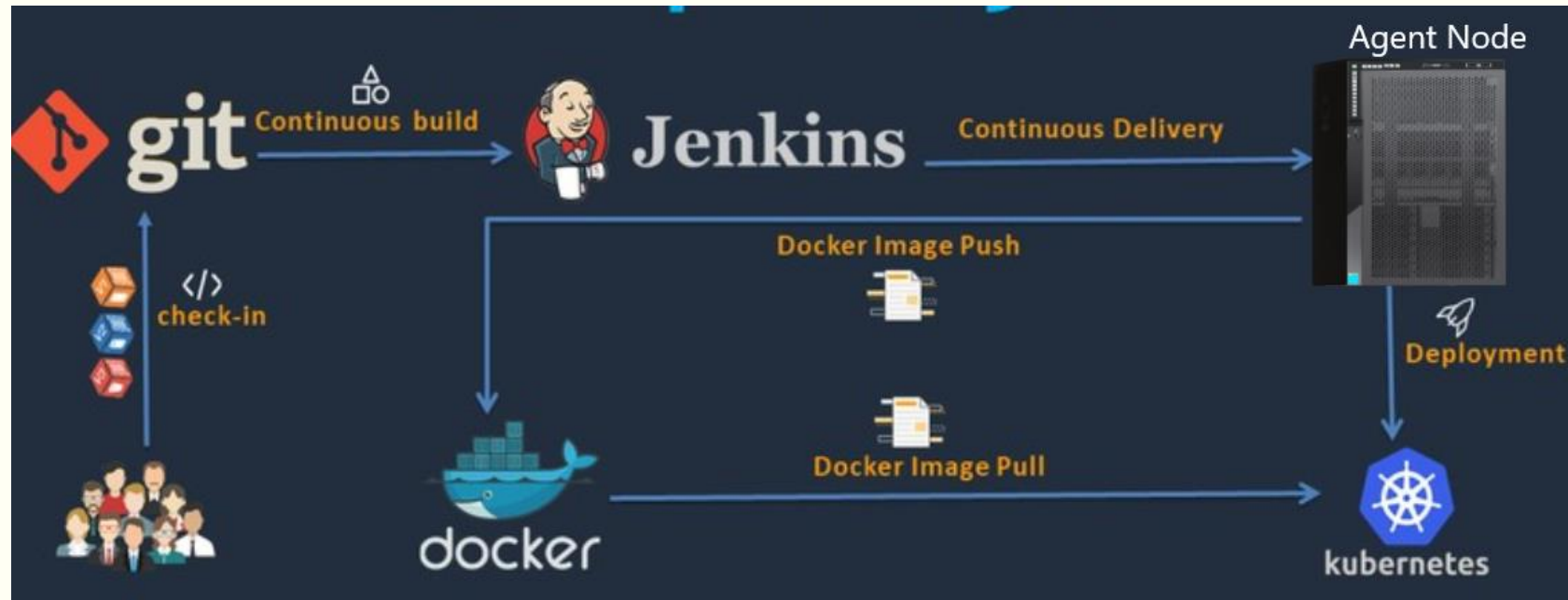
Refresh Filter -- Show/hide columns --

Displaying 1 - 8 out of 8 test run(s). Filtering results by Release. Clear Filters

✓	Name	End Date	Test Set	Release	Execution Status	Operating System	ID	Web Browser	Edit
<input type="checkbox"/>	Filter		-- Any --	1.1.0.0 - Lib	-- Any --	-- Any --	TR	-- Any --	Edit
<input type="checkbox"/>	Ability to create new author	14-Mar-2019		1.1.0.0.0002	Caution		TR:17		Edit
<input type="checkbox"/>	Ability to create new author	13-Mar-2019		1.1.0.0.0003	Caution		TR:20		Edit
<input type="checkbox"/>	Ability to edit existing book	11-Mar-2019		1.1.0.0.0003	Failed		TR:19		Edit
<input type="checkbox"/>	Ability to edit existing book	11-Mar-2019		1.1.0.0.0002	Blocked		TR:16		Edit
<input type="checkbox"/>	Ability to edit existing book	11-Mar-2019		1.1.0.0.0001	Passed		TR:14		Edit

01.09.2020

# Випуск. Контейнеризація програмного продукту



- Контейнер (екземпляр образу) Docker містить:
  - Обрана операційна система (наприклад, дистрибутив Linux, Windows Nano Server або Windows Server Core).
  - Файли, додані в ході розробки, наприклад первинний код та двійкові файли й додатки.
  - Дані щодо конфігурації, наприклад, параметри середовища та залежності.
- Компоненти Docker

# Випуск. Популярні графічні інтерфейси для Docker

## ■ Приклад: управління образами в Portainer

portainer.io

ACTIVE ENDPOINT  
primary

ENDPOINT ACTIONS

- Dashboard
- App Templates
- Services
- Containers
- Images**
- Networks
- Volumes
- Swarm

PORTAINER SETTINGS

- Password
- Endpoints

Portainer v1.11.3

Image list

Images

admin  
log out

Pull image

Name: e.g. ubuntu:trusty

Registry: leave empty to use DockerHub

Note: if you don't specify the tag in the image name, latest will be used.

Pull

Images

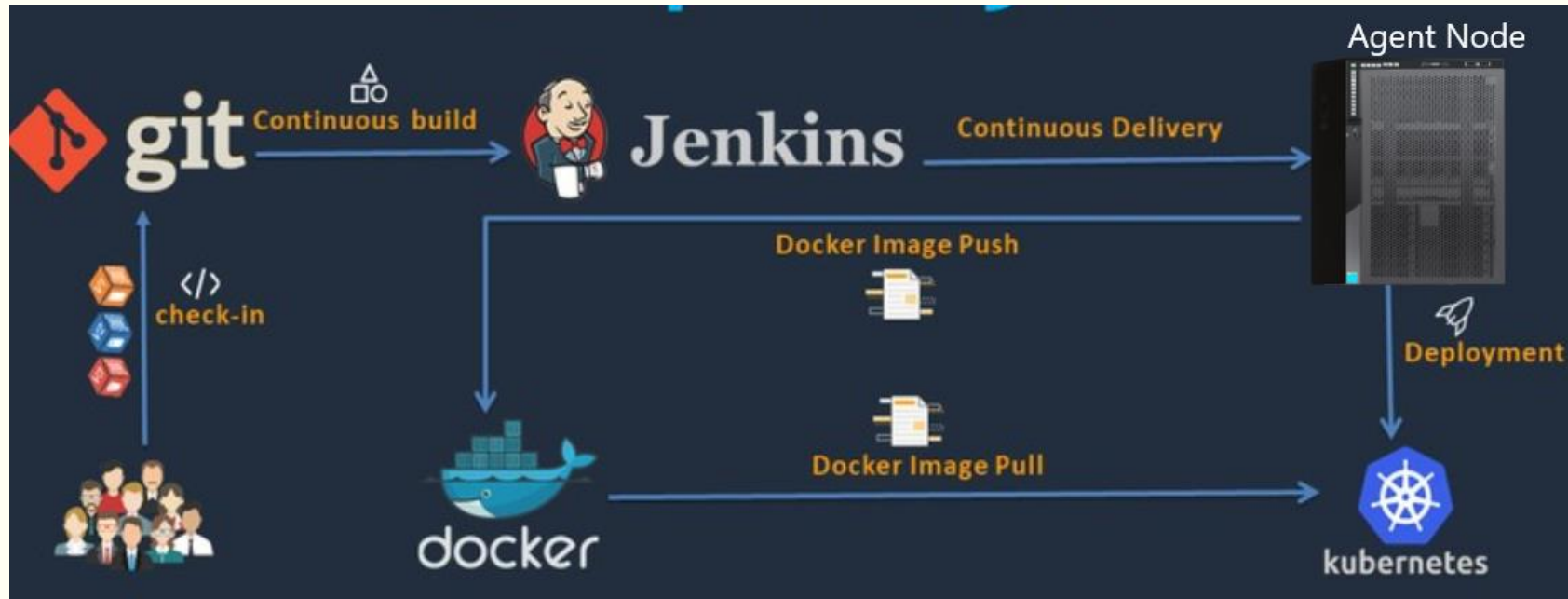
Items per page: 10

Remove

Filter...

Id	Tags	Size	Created
sha256:3ae7073f06...	wordpress:4.7.0	399.3 MB	2016-12-07 22:42:10
sha256:4956c56885...	ubuntu:yakkety	101.4 MB	2016-11-16 23:58:58
sha256:104bec311b...	ubuntu:latest	129 MB	2016-12-15 20:45:24
sha256:8b3c701ad4...	transcribator_web:latest	731.9 MB	2016-12-09 19:02:51
sha256:0adc78b813...	transcribator_nginx:latest	181.5 MB	2016-12-09 19:02:51
sha256:a399f36700...	python:3.5	682.5 MB	2016-11-10 11:26:04
sha256:95dfb9bfd6...	python:2.7	672.1 MB	2016-11-18 01:19:20
sha256:b06da01ac3...	postgres:9.6 postgres:latest	264.8 MB	2016-11-09 03:19:14

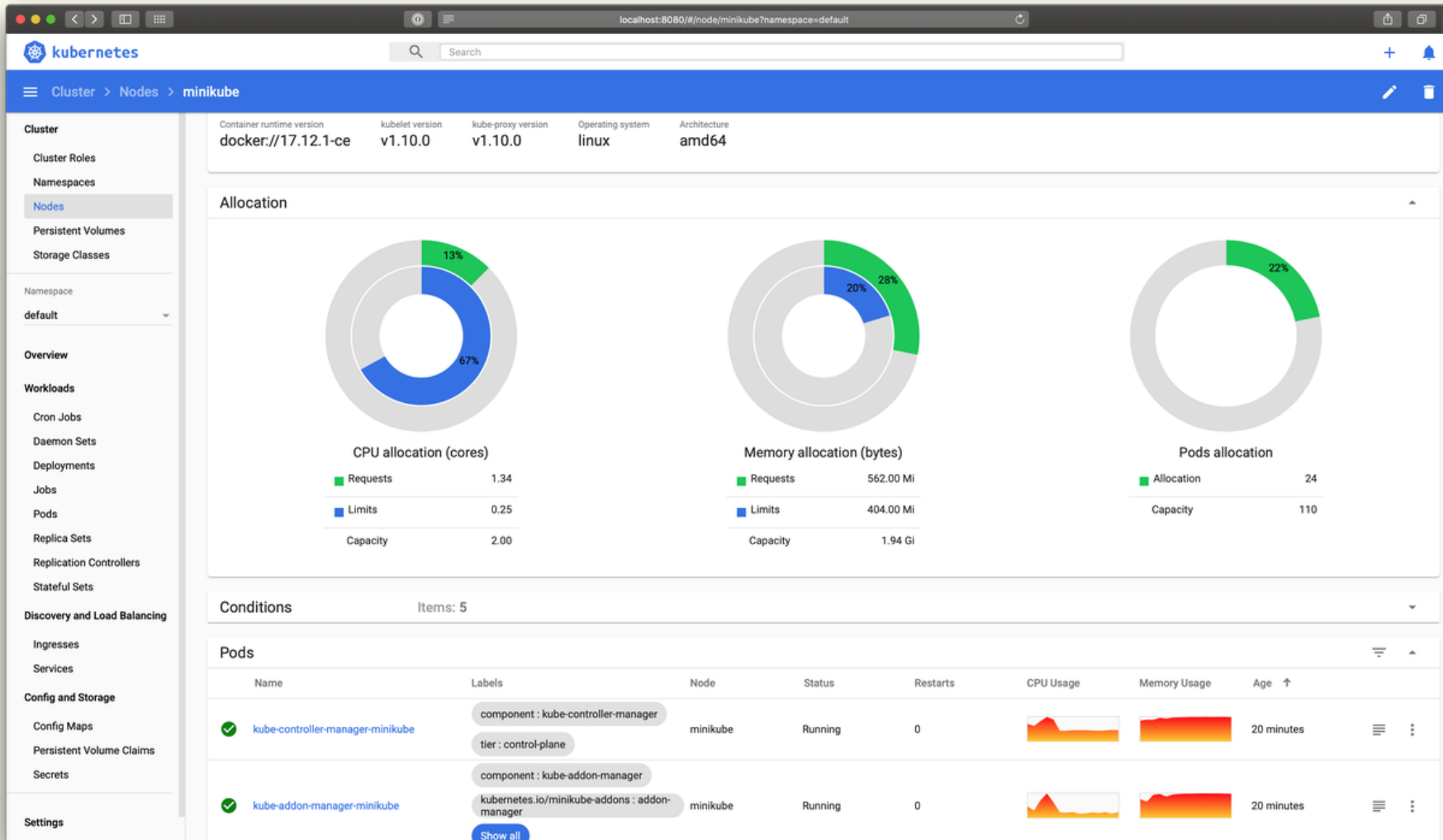
# Випуск. Розгортання (deployment) програмного продукту



- Kubernetes — це програмне забезпечення з відкритим кодом, яке надає API для керування місцем та способом виконання контейнерів.
  - Дозволяє запускати контейнери та робочі навантаження Docker, допомагає долати деякі складнощі при переміщенні для масштабування кількох контейнерів, розгорнутих на кількох серверах.
  - Kubernetes дозволяє координувати цілий кластер віртуальних машин та планувати запуск контейнерів на таких віртуальних машинах залежно від доступних обчислювальних ресурсів та потреб кожного контейнера в таких ресурсах.

# Випуск. Графічні інтерфейси для Kubernetes

- Приклад: веб-інтерфейс Kubernetes Dashboard







# ОПЕРАЦІЙНЕ ОБСЛУГОВУВАННЯ. МОНІТОРИНГ, ЛОГУВАННЯ ТА НАЛАГОДЖЕННЯ (KUBERNETES)



# ДЯКУЮ ЗА УВАГУ!

Наступна тема: Представлення програми в пам'яті комп'ютера