

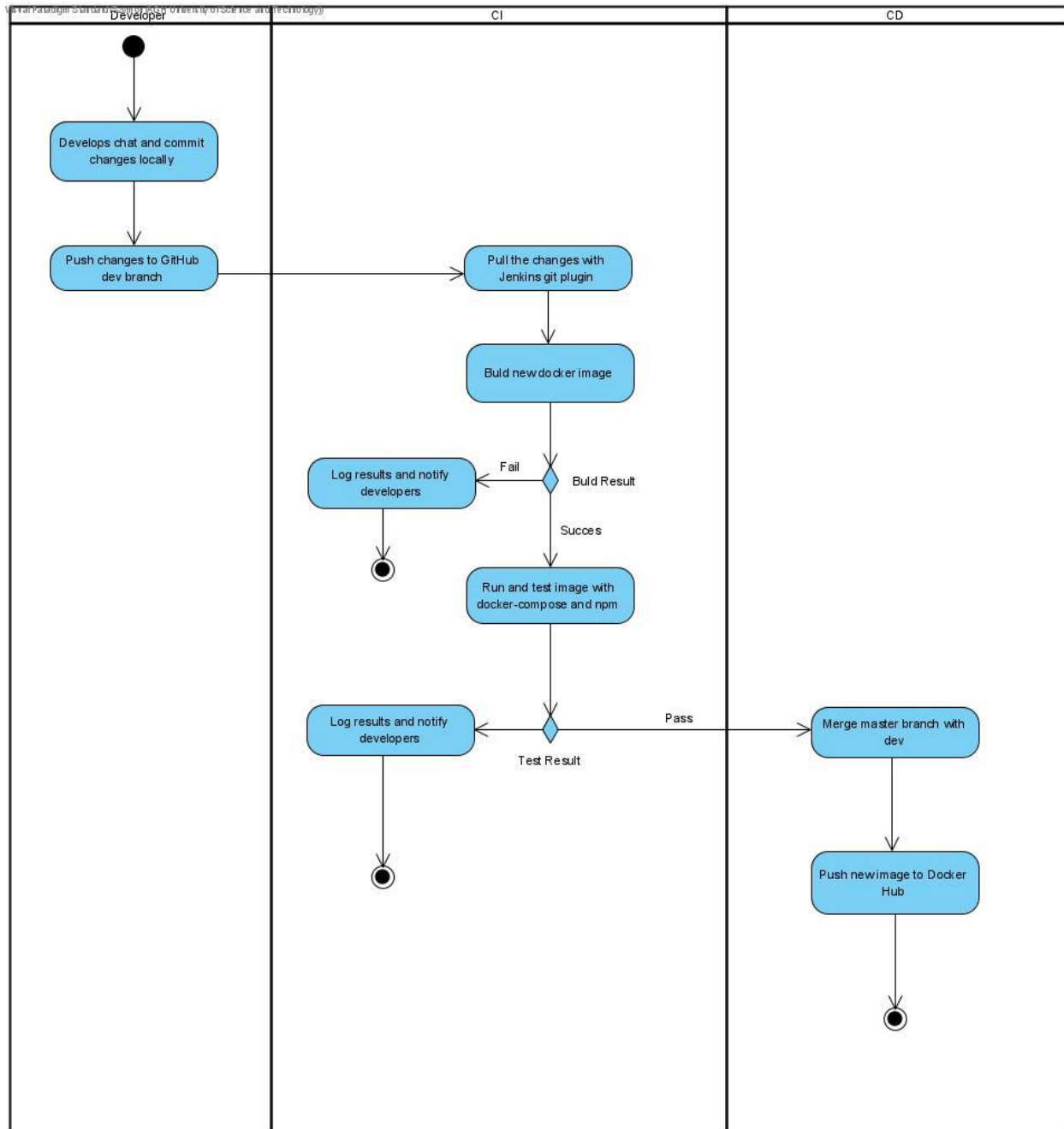
Socket.io-Messenger release pipeline plan

Author: **Szymon Karpecki**

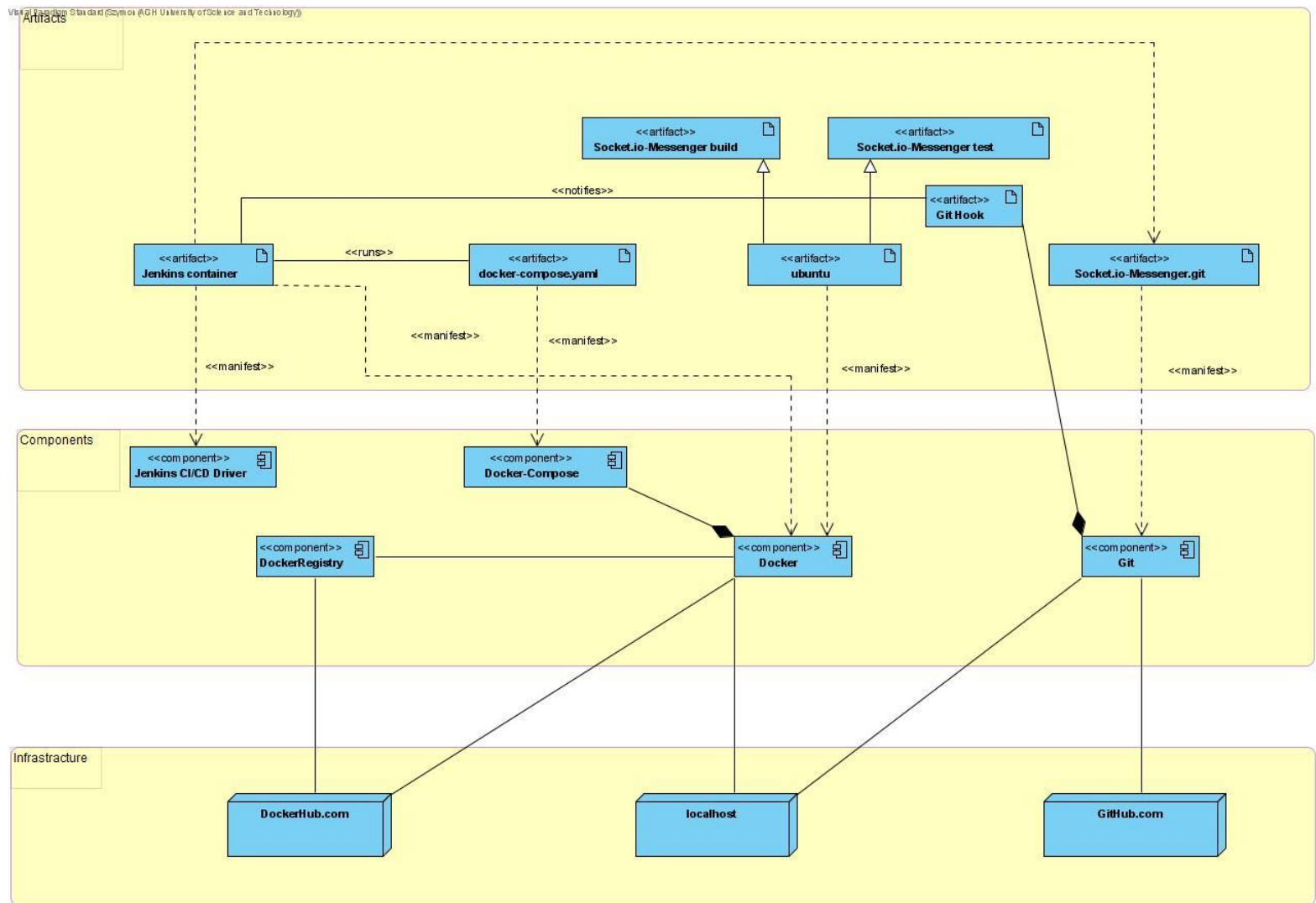
Technologies used:

- Git (GitHub)
- Jenkins
- Docker
- Docker-Compose
- DockerHub
- Node JS

1. UML Activity Diagram



2. UML Deployment Diagram



3. Description

Prerequisites:

1. Docker container built from Jenkins image (jenkins/jenkins) with docker-compose installed and pulled docker-compose and dockerfile files.
2. Second DIND container (docker:dind) which will be used to build and run application containers.
3. Configured Jenkins.

Integration:

1. When pushing changes to repository Git Hook sends API call to start Jenkins Job.
2. Jenkins pull the newest version of Socket.io-Messenger repo using Git plugin.
3. Run docker-compose which defines build and testing process in Jenkins container.
 - a. If building image succeeds go to tests.
 - b. Run tests with npm test in testing container.

Deployment

1. If tests were fine merge master with dev branch in Socket.io-Messenger repository.
2. Push newly build image to DockerHub