

## Творче завдання

1. Чи потрібна дана дисципліна?

Да, она нужна и полезна.

2. Якщо потрібна, то у такому ж форматі чи іншому?

Формат действительно лучше сменить, и дать это как сопровождение к какому то другому проекту, например у нас был проект где мы создавали веб приложение, сайт с базой данных, вот к этому проекту в сопровождение и гит и докер и ну даже дженкинс мог бы быть полезен, я лично гитом там и так пользовался. То есть давать этот предмет на том же семестре что и у нас была разработка того веб приложения.

3. Що змінити у темах дисципліни чи у переліку тем?

На самом деле все довольно хорошо, можно было бы несколько потрогать сами технологии. С учетом того что для докера, гита и иже с ними технологий линукс является родной системой, и на линуксе с ними проблем было практически 0, было бы неплохо сделать это дефолтной системой для изучения дисциплины. Возможно давать предмет **после** прохождения семестра курса операционных систем.

4. Які теми є зайвими?

5. Які теми розшити?

6. Які теми скоротити?

7. Які теми додати?

## На 4-7 я дам смешанный ответ

Тема про работу с облаком ощущается странно, она какбы и по теме но при этом ощущается так оторвано от основного курса, впрочем это дополнительное задание, но при этом мы почти весь семестр интегрировали изучаемые сервисы один с другим.

Мы вот уже выработали тот момент что удобнее запускать дженкинс на докере, возможно это все можно сократить, выделить больше времени на написание более сложных конвейеров, с помощью и скриптов и дженкинс файлов.

И выделить больше внимания работе с облаком, возможно я вот не сделал докер сварм и вот мне сейчас именно его нехватает, вот, не знаю.

## Які теми скоротити?

Еще есть вот темы про методологии разработки, scrum, eXtreme, все вот это, они в моем случае прошли почти мимо, к экзамену я повторял почти только по ним, потому что остальное все легко запомнилось. Как мне кажется тут скорее дело в самой структуре изучаемых вещей, это можно сократить до изучения только двух абстрактных тем, статичные vs гибкие, рассмотреть различия между ними, какие они вводят требования к организации рабочего процесса, и все, и не вкапываться в 50 видов каждой из методологий.

## Які теми додати?

Infrastructure as Code, с инструментом Terraform, они интегрируются с AWS и другими облаками, вроде как ползена и повсеместо используемая штука к AWS, что важно приносит вот этот концепт устройств как файлов конфигурации