Для початку, панове з techtarget.net [стверджують](https://www.techtarget.com/searchitoperations/definition/DevOps) що не існує жодного офіційного DevOps-фреймворку. Менше з тим, перелік знайдених не офіційних фреймворків:

1. [The Three Ways](https://blog.sonatype.com/principle-based-devops-frameworks-three-ways) - це те, що було описано в The Phoenix Project/DevOps Handbook. Складається з трьох “шляхів”:
   1. Принципу потоку - прозорість процесів, розбиття задач на маленькі складові, тісна комунікація між командами, оптимізація всього, що можливо;
   2. Принципу зворотнього зв’язку - можливість знайти та вирішити проблему якнайшвидше, закладання та підтримка високого рівня якості з самого початку розробки коду, оптимізація для наступних ланок процесу розробки;
   3. Принципу постійного навчання - введення культури роботи над помилками, оптимізація робочого процесу кожного члена команди, розповсюдження гарних ідей та знахідок між усіма членами команд, гнучкі шаблони щоденної організації роботи та акцент на культурі постійного розвитку кожного.

## [Technical and Management Practices of High-Performing DevOps Teams](https://blog.sonatype.com/principle-based-devops-frameworks-accelerate) від Accelerate. Складається з практик технічних та практик менеджменту.

Технічні практики описують ключові зони:

* 1. безперервна доставка, що власне описує методи CD;
  2. архітектура, де виділяють ідею архітектури слабкого зв’язку, тобто надання командам можливості довести свою роботу від початку до кінця без взаємодії з іншими командами та ідею уповноважених команд, тобто вибір командами інструментів самостійно для підвищення ефективності роботи;
  3. продукт та процеси де виділяють принципи зворотного зв’язку з клієнтом, ідеї, що оптимізація роботи приносить користь клієнту, роботи малими частками та право команд на експерименти.

Практики менеджменту ділять на дві частини - бережливе управління і моніторинг (перша частина) та культура.

Перша пропонує мінімізувати процес погодження змін, описує важливість постійного моніторингу, мінімізацію кількості проектів в роботі у однієї команди та важливість використання канбану чи якогось подібного методи візуалізації процесів.

Культура же пропагує навчання та саморозвиток, заохочує спільну роботу між командами, налаштування робочого процесу так, щоб він приносив працівникам задоволення та трансформаційне лідерство.

1. Власне, [CALMS](https://www.atlassian.com/devops/frameworks/calms-framework). Тобто Culture, Automation, Lean, Measurement, Share.
2. [CALMR](https://www.scaledagileframework.com/calmr/) від Scaled Agile. Розшифровується як Culture, Automation, Lean Flow, Measurements, Recovery. Є варіацією CALMS але відрізняється тим, що:
   1. ідеї спільних знань, досвіду та цілей тут відноситься до культури;
   2. у модуля Lean назва саме Lean Flow і він більше про слідування принципам канбан, ніж про творчі ідеї саморозвитку та оптимізації та прийняття помилок;
   3. реакція на помилки та проблеми у роботі винесена в окремий повноцінний модуль Recovery, який сфокусований навколо максимального скорочення часу простою, шляхом відпрацювання реакцій на можливі проблеми заздалегідь, вирішення них всіма учасниками процесу, як однією командою та впровадженням виправлень максимально швидко.

Якщо порівняти CALMS з, наприклад, The Three Ways, то можна помітити, що вони пропагують одні й ті самі цінності: тісну комунікацію між учасниками процесу, постійний розвиток, оптимізацію процесів та культуру організації робочих процесів в цілому. Тим не менше, в CALMS окремо виділяється і акцентується увага на, по-перше, автоматизації, як засобі оптимізації та підвищення якості, та, по-друге, на вимірюванні, тобто моніторингу та порівнянні отриманих результатів, як на одному з найнадійніших методів отримання зворотного зв’язку.