სოფტვეარ არქიტექტურა ქალაქის დაგეგმარებასავითაა. არქიტექტურის შექმნისას ჩვენ სისტემას ვყოფთ სხვადასხვა ნაწილებად და ვანიჭებთ მათ რაღაც ფუნქციონალს. ასეთი მოწყობა საშუალებას გვაძლევს რომ თუ ცვლილებები დაგვჭირდება მათი შესრულება დიდი პრობლემების გარეშე შევძლოთ. ცვლილებების ინიციატორი და გამომწვევი ბევრი მიზეზი შეიძლება იყოს, მაგალითად:

1. დამკვეთის მოთხოვნილი ახალი feature ის დამატება
2. სოფტის განახლება რათა ახალ რეგულაციებს შეესაბამებოდეს
3. სისტემის ნაწილების განახლება პერფორმანსის გასაუმჯობესებლად
4. დამოკიდებული სერვისების ცვლილების გამარტივბა
5. სულაც ინფრასტრუქტურის ცვლილება.

Hexagon, clean, onion და Layered არქიტექტურები სოფტის დალაგების სხვადასხვა გზებია. ისინი ყველა იყენებენ Separation Of Concern წესს. ეს წესი ნიშვნას იმას რომ აპლიკაცია უნდა დაიყონს რამდენიმე ნაწილად, დონედ და ყველა მათგანს უნდა ჰქონდეს განსაზღვრული/გაწერილი კონკრეტული დავალება.

ამ არქიტექტურებს შორის ძირითადი სხვაობა ისაა თუ როგორ განსაზღვრავენ საზღვრებს და როგორ აქვთ მოწყობილი სხვადასხვა მოიდულების ურთიერთობა]