[What is software development?](https://www.ibm.com/topics/software-development" \l "1309741)

1. Selecting a methodology (SDM; Software Development Methodology)
2. Gathering requirements
3. Choosing or building an architecture
4. Developing a design
5. Building a model
6. Constructing code
7. Testing
8. Managing configuration and defects
9. Deploying
10. ~~Migrating data~~
11. ~~Managing and measuring the project~~

+ optional

* [naming convention](https://cloud.google.com/apis/design/naming_convention)

Case1

[**V-model**](https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Systems_Engineering_Process_II.svg/1920px-Systems_Engineering_Process_II.svg.png)

1. Gathering requirements

* (Unknown)

1. Choosing or building an architecture

* Layered (n-tier) architecture: [**MVC**(Model, View, Controller/Presenter)](http://techbeacon.com/sites/default/files/fsgn4_1.png)

1. Developing a design

* [UI/UX design](https://yslab.kr/74) (optional)

1. Building a model

* UML: [Class Diagram](https://www.diagrams.net/assets/img/blog/class-diagram-example.png) / [Use case Diagram](https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Use_case_restaurant_model.svg/1024px-Use_case_restaurant_model.svg.png) / [ER Diagram](https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/ER_Diagram_MMORPG.png/673px-ER_Diagram_MMORPG.png) (optional)

1. Constructing code
2. Testing

* (optional)

1. Managing configuration and defects
2. Deploying

Case2

[**TDD(Test-Driven Development)**](https://gmlwjd9405.github.io/2018/06/03/agile-tdd.html)

1. Gathering requirements

* (Unknown)

1. Choosing or building an architecture

* Layered (n-tier) architecture: [**MVC**(Model, View, Controller/Presenter)](http://techbeacon.com/sites/default/files/fsgn4_1.png)

1. Developing a design

* [UI/UX design](https://yslab.kr/74) (optional)

1. Building a model

* UML: [Class Diagram](https://www.diagrams.net/assets/img/blog/class-diagram-example.png) / [Use case Diagram](https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Use_case_restaurant_model.svg/1024px-Use_case_restaurant_model.svg.png) / [ER Diagram](https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/ER_Diagram_MMORPG.png/673px-ER_Diagram_MMORPG.png) (optional)

1. Constructing code ***based on Unit Tests***
2. Deploying

클린 아키텍쳐

앱-티튜드 테스트: 먼저 동작하게 만든다. 그리고 올바르게 만든다. 그리고 빠르게 만든다. (P.271)

* 성능 측정 지표: 테스트 소요 시간,

Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Parameter | Name | Include | Return |
| Refactoring | TYPE.FUNCTION( any ) | calcExecutionSeconds | Performance.now | TYPE.INT( time.seconds ) |
| Testing |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

reference:

<https://www.ibm.com/topics/software-development#1309741>

<https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice>

<https://gmlwjd9405.github.io/2018/07/04/class-diagram.html>

<https://gmlwjd9405.github.io/2018/06/03/agile-tdd.html>

<https://cloud.google.com/apis/design/naming_convention>

<https://yslab.kr/74>

<https://en.wikipedia.org/wiki/Method_stub>

[Test Driven Development By Example - Kent Beck.pdf](https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnx0ZXN0MTIzNHNpbTQ2NXxneDpiYTJmYWIwYTAyOGJiZmQ)

<https://github.com/serodriguez68/clean-architecture>

<https://github.com/piloulac/clean-code>

수정사항:

내용 추가 (2022-09-01)