

1. SW 개발 프로세스
  - 1-1. Waterfall
  - 1-2. Iterative - Agile process
2. 수강 신청 시스템
  - 2-1. 요구 사항
3. Chat GPT가 작성한 MVC 패턴

## 1. SW 개발 프로세스 (Waterfall, Agile)

프로세스는 공정. 업무를 처리하는 공정. 어떠한 업무의 단위를 뜯어서 시간적 순서대로 나열한 것을 말한다. 일반화시켜서 규칙을 만든 것이다. 실행 중인 프로그램을 프로세스라고도 한다. 수업에서 말하는 SW 개발 프로세스는 소프트웨어 개발자가 해야 하는 일의 순서이다. 대상이 CPU가 아니라 소프트웨어 개발자이다. 일반적으로 우리는 고객이 있어야 하고, 고객이 요구 사항을 가지고 있다. 보통은 고객 자신이 무엇을 원하는지 잘 모르기 때문에 개발자는 요구 사항을 소프트웨어 시스템으로 바꿀 수 있어야 한다.

기술을 가진 사람이 요구 사항을 기술적인 스펙으로 바꿔야 한다. 그러기 위해서는 요구 사항을 명확하게 분석해야 한다. 분석은 요구 사항을 정리하는 것이다. 그것을 기술적 사양으로 설계한다. 구현한 뒤에는 테스트를 거친다.

고객의 요구 사항을 가지고 고객의 생각을 정리한 다음에 그것을 실제로 설계해서 만드는 것이다. 이것이 소프트웨어 개발 프로세스이다.

이 안에는 요구 사항을 추출하는 과정이 포함되어 있다. 분석하기 전에 부족한 정보를 채워야 한다. 요구 사항조차도 개발자가 물어봐야 한다. 일반적으로 소프트웨어 개발 프로세스는 요구 사항을 추출하고, 그것을 분석하고, 설계하고, 구현하고, 확인하는 5단계로 구성된다.

요구 사항과 분석 작업에서는 고객의 관점에서 정리하는 것이고, 설계는 엔지니어의 관점에서 정리하는 것이다.

관점은 고객에서 개발자로 변하는데, 제공되는 서비스의 관점을 소프트웨어 기술 관점으로 바꾼다. 분석에서 해야 할 일 중 하나는 요약이다. 요약은 수많은 단어 중에서 골라서 짧게 만드는 일이다. 그리고 그 안에 큰 흐름이 들어가야 한다. 요약은 핵심 구성요소를 꺼낼 것인지, 순서를 고를 것인지로 나뉜다. 즉 명사로 고를 것인지 동사로 고를 것인지를 생각해야 한다. 구성요소로 요약을 한다는 것은 구성요소와 관계를 만들어내는 것이다. 순서로 요약을 한다면 시간의 흐름, 사건의 순서를 얘기하는 것이다. 분석하기 위해서는 이처럼 나름의 원칙이 있어야 한다.

객체 지향 프로그래밍은 분석을 객체(명사)와 관계 중심으로 한다. 그런데 소프트웨어 개발

프로세스는 5단계의 시간의 흐름으로 사건을 정리한 것이다. 무언가를 정리할 때에는 관점과 원칙이 있어야 한다.

정리한 요구 사항을 기술적 관점에서 또 정리해서, 어떤 기술적 요소(web, DB 등)로 만들 것인지에 대한 것을 설계단계에서 한다. 그리고 설계된 것을 기계가 이해할 수 있는 언어로 만들어낸다. 테스트할 때는 의도한 바와 결과의 차이를 가지고 바라봐야 한다. 즉 만들어진 것을 확인할 때에는 요구 사항과 실행되는 결과를 비교해봐야 한다.

소프트웨어 개발 프로세스에서는 초기에는 기능이 추가되다가 과정이 진행될수록 수정이 추가된다.

### 1-1. Waterfull

Waterfull 프로세스는 초반부터 요구 사항을 정밀하게 파악하고 분석하고 설계하고 구현해서 마치 폭포처럼 모든 단계를 한 번에 끝내는 것을 말한다. 다시 돌아가지 않는다는 특징이 있다.

하지만 요즘에는 대부분 반복적(Iterative) 프로세스를 쓴다. 그러나 반복적 프로세스는 계속 반복되기 때문에 진행 상황을 정확히 알 수 없어서 문제가 생겼을 때 귀책사유를 묻기 어렵다는 단점이 있다. 즉 반복적 프로세스의 핵심은 고객이 구현한 것을 봐야 수정해야 할 것을 알 수 있다는 것이다. 그래서 반복적 프로세스는 시간을 정해놓고 한다.

### 1-2. Iterative - Agile process

가장 많이 쓰는 반복적(Iterative) 프로세스는 Agile 프로세스이다. 이 프로세스는 내가 해야 할 일을 먼저 적어놓고 진행하는데, 계속 요구 사항이 반복적으로 추가되고 수정된다.

내가 아는 범위 안에서 해볼 수 있는 것을 해보면, 무엇이 더 필요한지 알 수 있게 된다. 즉 할 일을 먼저 적어놓고 진행하는 Agile 프로세스는 내가 할 수 있는 범위 내에서 진행하는 것과 같다.

“추가 요구 사항의 수용 측면에 있어서 Agile process는 처음 수집한 요구 사항을 전체 요구 사항 중 일부로 인정하고 시작하므로 언제든지 추가 요구 사항이 있을 것으로 간주한다. 따라서 추가 요구 사항을 수용할 수 있는 방법으로 설계되어 있다. Waterfull 모델은 요구 사항 분석이 완전히 완료된 후에 설계 단계로 넘어가므로 새로운 요구 사항을 추가하기 쉽지 않다. 추가 요구 사항을 반영하기 어려운 구조이다.”[네이버 지식백과] 애자일 방법론과 폭포수 모델의 비교 (쉽게 배우는 소프트웨어 공학, 2015. 11. 30., 김치수)

## 2. 수강 신청 시스템

우리가 만들 수강신청 시스템은 Enterprise Application, 혹은 IT Service이다. Enterprise Application은 1대 n의 관계, 서비스 시스템을 말한다. 클라이언트와 서비스를 제공하는 서버

시스템이 있다. n명의 클라이언트가 하나의 서버를 공유한다. 서로 같이 써야 할 자원이 서버 있기 때문에 공유하는 것이다. Enterprise는 무언가를 공유하는 시스템이다.

기술, 부품이 표준화되듯이 소프트웨어도 표준화되기 시작했다. 그러기 위해서는 아키텍처가 표준화되어야 한다. 아키텍처는 시스템을 만들 때 시스템을 구성하는 가장 근본적인 구성을 요약해놓은 것이다. 요약은 어떤 관점에서 해야 하는 지가 중요하다.

소프트웨어 아키텍처를 구조적인 관점에서 보면, Enterprise Application 패턴이 나타난다. 패턴은 공통으로 쓰는 구조의 연결성을 말한다. 우리가 쓸 패턴은 Model - View - Controller 세 개의 계층으로 나눈다. 이를 MVC 패턴이라고 한다.

“사용자의 입력이 컨트롤러로 들어오면 컨트롤러는 모델에서 데이터를 불러오고, 해당 데이터를 뷰를 통해 화면으로 출력한다. 모델, 뷰, 컨트롤러가 기능별로 독립적으로 분리되어 있어서 여러 명의 개발자가 동시에 각각의 기능을 개발할 수 있고, 일부 기능이 추가될 경우도 기존의 구성 요소를 재사용할 수 있어 코드 재사용성이 향상된다.”[네이버 지식백과] 모델 · 뷰 · 컨트롤러 [Model-View-Controller] (IT용어사전, 한국정보통신기술협회)

View는 사용자가 컴퓨터로 연결되는 오감에 관련되어 있다. 웹페이지나 UI를 말한다. Model은 지속적 데이터 관리를 담당한다. 한글이나 파워포인트가 저장될 때 언어적으로 정리를 해놓는 것을 말한다. 우리가 직접 보는 형태와는 다른 형태로 저장된다. 대규모 시스템의 경우 모델에 데이터베이스가 붙어있고, 아니면 파일이 붙어있다. 지속적 데이터를 관리하는 것이 파일 시스템이다.

데이터베이스에 저장된 데이터와 내가 View에서 보는 데이터는 차이가 난다. 그래서 중간에 실제 데이터와 화면(View)을 연결해주는 중간 작업자(관리자, controller)가 있어야 한다.

## 2-1. 요구 사항

Story/Usecase/Scenario는 요구 사항, 기능, 사용자 시나리오를 말한다.

### 1. 로그인

- 아이디와 패스워드를 모두 입력하여 로그인한다.
- 아이디나 패스워드가 일치하지 않는 경우 “아이디 또는 패스워드가 틀렸습니다”라는 메시지를 보여준다.
- 계정이 아예 존재하지 않는 경우 “계정이 존재하지 않습니다”라는 메시지를 보여준다.
- 5번 이상 연속으로 로그인에 실패할 경우, 그 사용자는 1시간 동안 로그인이 불가능하다.

### 2. 강좌 선택

- 로그인된 사용자만 강좌를 선택할 수 있다.
- 사용자는 자신이 원하는 강좌를 선택할 수 있다.
- 강좌를 선택할 때에는 중복되지 않아야 한다.

### 3. 미리 담기

- 미리 담은 강좌는 장바구니 페이지에 저장된다.
- 미리 담은 강좌는 수강신청 페이지 진입 후 사용자가 쉽게 볼 수 있고 접근할 수 있어야 한다.
- 사용자는 이미 선택한 강좌를 '미리 담기' 할 수 있다.

### 4. 수강 신청

- 로그인된 사용자만 수강신청 할 수 있다.
- 수강신청은 중복되지 않아야 한다.
- 수강 정원이 초과된 강좌는 "수강 정원이 초과되었습니다."라는 메시지를 보여주고, 수강신청을 할 수 없다.

### 5. 확인

- 사용자는 수강 신청한 강좌 목록을 수강 신청 후 바로 확인할 수 있다.

요구 사항은 반복될수록 구체화 되고 추가된다.

## 3. Chat GPT가 작성한 MVC 패턴



MVC (Model-View-Controller) 패턴은 소프트웨어 개발에서 사용되는 소프트웨어 디자인 패턴 중 하나로, 애플리케이션의 구성 요소들을 세 가지 역할로 나누어 개발하는 방법론입니다.

1. 모델(Model): 데이터와 비즈니스 로직을 담당합니다. 데이터를 관리하고, 데이터에 대한 조작, 처리, 저장 등의 기능을 수행합니다.
2. 뷰(View): 사용자에게 보여지는 인터페이스를 담당합니다. 데이터를 시각적으로 표현하고, 사용자의 입력을 받아 컨트롤러로 전달합니다.
3. 컨트롤러(Controller): 모델과 뷰를 연결하는 다리 역할을 수행합니다. 사용자의 입력을 받아 모델의 데이터를 갱신하고, 뷰를 업데이트하여 사용자에게 결과를 표시합니다.

MVC 패턴은 각 구성 요소들이 독립적으로 동작하며, 각각의 역할에만 집중하도록 설계되어 애플리케이션의 유지 보수성과 확장성을 높입니다. 또한, 사용자 인터페이스와 비즈니스 로직을 분리하여 개발하므로 코드의 재사용성이 높아집니다.

