

목차

1. 개발 프로세스

A. Water Fall

B. Iterative

i. Agile

2. Functional View

A. Domain Knowledge

3. Object-Oriented View

A. 명사

B. 동사

C. 형용사

D. 부사

1. 개발 프로세스

개발 프로세스를 보면 요구사항을 수집해서 정리를 한다. 그리고 정리한 내용을 바탕으로 분석을 하고 설계를 한다. 설계한 내용을 바탕으로 구현을 하고 구현한 것을 테스트 한다. 이것을 액티비티라 하고 이것이 프로세스를 구성하는 행위다.

개발 프로세스는 크게 waterfall 과 iterative 로 나눌 수 있다.

A. Water Fall

Waterfall 은 모든 요구사항을 한 번에 잘 정리하고, 한 번에 잘 설계해서 한 번에 끝내 버리는 것이다.

B. Iterative

i. Agile

Agile 은 반복적 프로세스 중에 가장 많이 쓰는 프로세스다. 반복적 프로세스는 업무를 반복적으로 나눠서 하는 방법이다.

2. Functional View

요구사항은 사용자의 관점, 설계는 개발자 관점, 분석은 도메인 엑스퍼트와 사용자 관점이다. 분석은 업무의 내용을 알아야 하는데 업무에는 domain knowledge 를 고집어내야 한다.

A. ¹Domain Knowledge

Domain knowledge 는 인간활동의 영역이나 자율적인 컴퓨터 화동, 다른 전문 분야에서 사용되어지는 유효한 지식으로, 업무를 분석하는 것이다.

도메인은 특정 영역의 업무를 뜻한다. 유저가 요구를 하면 도메인 엑스퍼트 그걸 구체적으로 정리를 한다.

¹ <https://okky.kr/articles/520776>

고객은 우리 시스템을 서비스 제공자로 본다. 시스템은 인위적으로 사람이 만든 것이다. 시스템은 하나 이상의 무언가 모여서 어떠한 목적을 가지고 움직인다. 그리고 고객에게 제공해준다. 제공한다는 것은 입력을 주면 아웃풋을 주는 것이다.

함수는 원소끼리 대응관계를 규칙을 만들어 놓은 것이다. 제약조건은 어떤 입력이 들어와도 확정적 결과를 가져와야 한다. 함수를 사용하는 이유는 예측하기 위해서다. 미리 어떤 행위가 일어날지 규칙을 파악하기 위해서다.

또한 함수라는 것은 데이터 관점으로 볼지 규칙 관점으로 볼지에 따라 다르다. 함수를 다른 말로 표현하면 데이터플로우다이아그램으로 표현할 수 있다. 규칙을 흐름으로 표현해 놓은 것이 데이터플로우다이아그램이다. 함수가 어떠한 인풋을 받고 어떤 아웃풋을 가지는 지를 표현하는 것이다.

3. Object-Oriented View

객체 관점은 함수보다 데이터가 더 중요하다. 데이터 중심으로 함수의 집합을 객체로 만드는 경우도 있다.

A. 명사

명사는 시간이 지나도 변하지 않는 것이다. 구조와 같다.

B. 동사

동사는 함수로, 시간이 지남에 따라 주체의 변화에 관한 것이다. 행위에 관한 것이므로 주체가 반드시 있어야 한다.

C. 형용사

상태를 표현한다. 따라서 특정 시점에서 명사가 가진 속성의 집합을 의미한다.

D. 부사

동사가 어떻게 실행되어야 하는 지, 즉 동사를 제약한다. 동사의 구체적인 내용에 관한 이야기, 동사가 어떻게 변해야 하는지에 대해 설명한다. 그리고 규칙을 실행하는 방법을 제약하는 것이다.