

제1장 소프트웨어공학 시작

1. 개요

먼저, IEEE에서는 '소프트웨어공학'을 다음과 같이 정의하고 있다.

—〈소프트웨어공학 정의〉—

소프트웨어공학은 '소프트웨어의 개발, 운용, 유지, 파기에 대한 체계적인 접근 방법으로 공학을 소프트웨어에 적용한 것이다'

—〈IEEE〉—

◆ 소프트웨어공학 목표

- ① 좋은 품질의 소프트웨어를
- ② 계획된 최소의 예산(비용)으로
- ③ 주어진 개발 일정에 맞추어 완성한다.

◆ 소프트웨어 위기

소프트웨어 공학 탄생 원인	<ul style="list-style-type: none"> • 소프트웨어 공학의 용어는 1968년 나토 소프트웨어 공학 학회에서 사용 • 당시, 소프트웨어 위기에 대해 사람들이 관심을 가지도록 장려하기 위해서다 • 소프트웨어가 하드웨어 개발 속도를 따라가지 못함 • 소프트웨어가 더 이상 사용자들의 요구를 충족시킬 수 없음
----------------	--

원인	<ul style="list-style-type: none"> • 소프트웨어 특성에 대한 이해 부족 및 관리 방법론 부재 • 소프트웨어 개발에 대한 전문적 교육 부족 • 작업일정과 비용의 추정치가 부정확
결과	<ul style="list-style-type: none"> • 개발 인력 부족과 인건비 상승(개발기간 및 비용 증가) • 소프트웨어 성능 및 신뢰성 부족(소프트웨어 생산성 저하) • 소프트웨어 품질저하 및 유지보수 비용 증가
극복 방안	<ul style="list-style-type: none"> • 자동화 : 형상관리, CASE, UML/ERD • 품질보증 : CMMi, SPICE • 공학적 접근 : 구조적 방법, 정보공학 방법, 객체지향 방법, CBD 방법 • 표준화 : IEEE, ISO, IEC, IEFT, W3C

(1) 소프트웨어 특징

비가시성(invisibility)	소프트웨어 구조가 외부에 노출되지 않고 코드에 숨어 있다.
무형성(intangible)	실제 형체가 없다.(기능점수 등으로 유형화 한다)
유형성(tangible)	분석, 설계사양서, 원시코드 등의 산출물을 볼 수 있다.
변경성(changeability)	요구사항 변경으로 인한 잦은 변경(필요에 따라 항상 수정 가능)
견고성(solidity)	구조 변경이 어렵다.(일부 수정은 전체에 영향을 줄 수 있다)
순응성(conformity)	사용자 요구나 외부 환경변화에 따라 적절히 변형 가능하다.
비마모성(longevity)	외부 환경에 마모되지 않는다.(단, 품질이 나빠질 수 있다)
복잡성(complexity)	개발과정이 복잡하고 비규칙이므로 이해가 어렵다.(비정형적)
복제성(duplicability)	적은 비용 또는 무상으로 간단하고 쉽게 복제가 가능하다.
동적행위성(dynamic activity)	하드웨어에서 동작하는 프로그램이다.
비제조성(개발)	제조(조립)가 아닌 개발이다.(developed not manufactured)
상품성(commercial value)	개발된 소프트웨어는 상품으로 판매된다.

(2) 소프트웨어 분류

소프트웨어는 시스템 소프트웨어와 응용 소프트웨어로 분류할 수 있다.

◆ 최근 공무원 시험에서 출제되고 있는 분류 방식

시스템 소프트웨어	• 시스템 소프트웨어는 사용자의 편리한 컴퓨터 사용을 위한 것(지원)	
	운영체제	Unix, Linux, Xenix, MS-DOS, Windows, OS/2, RTOS 등
	언어번역	컴파일러, 어셈블러, 인터프리터, 로더(loader), 링커(linker)
	유틸리티	시스템관리, 파일관리, 통신, 압축, 백신 프로그램
응용 소프트웨어	• 응용 소프트웨어는 사용자의 특정 작업 수행을 위한 것(목적 달성용)	
	범 용	문서편집, 스프레드시트(표계산), 자료관리, 프리젠테이션, 전자출판, 도형 편집 등 일반 업무를 처리하기 위해 사용되는 것
	특수용	금융, 부동산, 의료, 통계, 숙박, 음식, 운수, 개인관리, 시뮬레이션 등 특정 분야의 업무처리를 수행하기 위한 응용 프로그램

◆ 구매 방법에 따른 소프트웨어 분류

① 프리웨어(freeware)

- 프리웨어(freeware)는 **무료** 소프트웨어이다.
- 프리웨어는 계속 사용하여도 만든 사람이 대가를 바라지 않는 소프트웨어이다.
- 프리웨어는 사용자 집단에 따라서 프리웨어일 수도 있고 아닐 수도 있다.
- 개인 사용자에게는 무료이나 상업적으로 쓸 수 없는 소프트웨어도 있다.
- 프리웨어의 조건을 만족하지만 재배포가 불가능한 프리웨어도 있다.
- 프리웨어 정의는 명확하지 않다.
- 프리웨어 정의에 따라 프리웨어인가? 아닌가? 는 달라질 수 있다.
- 보통, 프리웨어라고 하면 대가 없이 사용할 수 있는 독점 소프트웨어를 가리킨다.
- 우리나라의 대표적인 프리웨어로는 **곰플레이어, 반디집, 알집**(개인에게는 무료) 등이 있다.
- 프리웨어는 무료로 사용할 수 있지만, 다른 권리까지 포기한 소프트웨어는 아니다.

② 셰어웨어(shareware) - 시험판, 평가판, 체험판, 후불소프트웨어

- 셰어웨어는 처음에는 공짜로 쓸 수 있다.
- 하지만, 셰어웨어는 기능 제한을 풀거나 **일정기간**이 지나 사용하려면 대가를 지불해야 한다.
- 셰어웨어는 처음에 무료로 제공한 후 유용성에 따라서 구매하도록 하는 소프트웨어이다.
- 셰어웨어는 "**구매하기 전에 한 번 써 보라**"라는 뜻을 내포하고 있다.
- 셰어웨어는 처음부터 유료로 판매하는 것을 목적으로 개발된 제품이다.
- 셰어웨어는 시험판이지만 전체 기능을 가진 채로 처음에 무료 배포되는 소프트웨어이다.

③ 라이트웨어(liteware)

- 라이트웨어는 상용 소프트웨어의 **일부 기능만** 사용 가능한 무료 배포되는 소프트웨어이다.
- 라이트웨어는 상용 소프트웨어에서 몇 가지 핵심 기능을 제거한 채 무료로 배포한다.
- 라이트웨어도 일종의 셰어웨어라고 간주될 수 있다.
- 라이트웨어와 셰어웨어는 사용해보고 마음에 들면 돈을 내고 구입하라는 것이다.

④ 오픈소스 소프트웨어(open source software)

- 오픈소스 소프트웨어는 **소스코드가 공개(open)**된 소프트웨어이다.
- 오픈소스 소프트웨어는 간단하게 오픈소스라고 지칭하기도 한다.
- 대부분의 오픈소스는 무료로 사용 가능하다.
- 참고로, 오픈소스이지만 사용하려면 돈을 내야 하는 것도 있다.
- 오픈소스는 재배포 및 개조가 가능하다.
- 오픈소스 이념은 소프트웨어는 한 사람보다는 여러 사람들이 개발하는 것이 좋다는 것이다.
- 대표적인 오픈소스로 서버 운영체제인 **리눅스**가 있다.
- 리눅스는 핀란드 헬싱키 공대학생이었던 리누스 토르발스가 만들었다.(1991년)
- 초기 리눅스는 기능이 불완전한 운영체제이었다.
- 하지만, 리눅스는 공개 직후부터 폭발적인 인기와 성장세를 보였다.
- 리눅스는 여러 단체들이 관심을 가지면서 완전한 운영체제로 거듭났다.
- 현재, 리눅스는 웹서버와 모바일 기기(안드로이드 등)를 구동하는 운영체제이다.
- 오픈소스는 독점적인 권리가 없을 뿐이지, 저작권은 존재한다.

기출문제 분석

1. 소프트웨어 개발 계획 단계에서 도출해야 할 중요한 2가지 요소로 가장 옳은 것은? [2018년 서울 7급]

- ① 비용과 일정 ② 시간과 일정
- ③ 자원과 도구 ④ 비용과 자원

☞ 소프트웨어 개발 계획 단계

- ① 좋은 품질의 소프트웨어를
- ② 계획된 최소의 **예산(비용)**으로
- ③ 주어진 **개발 일정**에 맞추어 완성한다.

정답 : ①

2. Lehman의 소프트웨어 진화 법칙에 대한 설명으로 가장 옳지 않은 것은? [2021년 서울 7급]

- ① 시스템이 릴리스된 후 그 시스템이 다른 시스템으로 대체될 때까지 변경된다.
- ② 진화하는 시스템의 유지보수 프로세스의 평균 작업량은 시스템이 소멸될 때까지 일정하다.
- ③ 소프트웨어 시스템의 구조는 변경될수록 복잡도가 낮아지는 경향이 있다.
- ④ 시스템의 평균 성장률은 소멸될 때까지 일정하다. 즉 시스템 개발의 전 단계에 걸쳐 각 버전의 변화는 거의 일정하다.

☞ 리먼(Lehman)의 소프트웨어 진화 법칙

- 소프트웨어 시스템의 구조는 변경될수록 복잡도가 낮아지는 경향이 있다.(x)
 - 소프트웨어 시스템의 구조는 변경될수록 복잡도가 높아지는 경향이 있다.
 - 소프트웨어 시스템의 구조는 변경이 가해질수록 구조는 복잡해진다.

// 리먼(Lehman)의 소프트웨어 진화 법칙

- 소프트웨어는 지속적으로 변화한다는 것을 전제로 시스템 변화에 대해 적용되는 원칙 제시

계속적 변경	소프트웨어는 계속 진화하며 요구사항에 의해 계속 변경
복잡도 증가	소프트웨어는 변경이 가해질수록 구조는 복잡해진다.
품질 저하	소프트웨어는 변화가 지속되면서 기능은 증가하지만 품질은 저하된다.

정답 : ③

5. 구매 방법에 따른 소프트웨어 분류에 대한 설명으로 옳은 것은? [2019년 지방 9급]

- ① 프리웨어(freeware)는 라이선스 없이 무료로 배포되어, 영리목적 기관에서도 자유롭게 배포할 수 있는 소프트웨어이다.
- ② 라이트웨어(liteware)는 상용 소프트웨어의 일부 기능만을 사용할 수 있도록 하여, 낮은 가격에 판매되는 소프트웨어이다.
- ③ 오픈소스 소프트웨어(open source software)는 프로그램 소스가 공개되어 있으나, 저작권자의 동의 없는 임의 수정은 불가능하다.
- ④ 셰어웨어(shareware)는 시범적으로 사용자에게 무료로 제공한 후 일정기간이 지나면, 유용성에 따라서 구매하도록 하는 소프트웨어이다.

☞ 구매 방법에 따른 소프트웨어 분류

- ① 프리웨어(freeware)는 라이선스 없이 무료로 배포되어, 영리목적 기관에서도 자유롭게 배포할 수 있는 소프트웨어이다.(×) → 영리목적 기관에서 자유롭게 배포할 수 없다.
 - ② 라이트웨어(liteware)는 상용 소프트웨어의 일부 기능만을 사용할 수 있도록 하여, 낮은 가격에 판매되는 소프트웨어이다.(×)
→ 라이트웨어는 사용해보고 마음에 들면 돈을 내고 구입하라는 것이다.
 - ③ 오픈소스 소프트웨어(open source software)는 프로그램 소스가 공개되어 있으나, 저작권자의 동의 없는 임의 수정은 불가능하다.(×)
→ 오픈소스는 재배포 및 개조가 가능하다.
-

정답 : ④