

- 품질 보증 : 개발되고 있는 소프트웨어가 요구와 품질 수준을 만족시킬 것이라는 것을 보장하는 작업
- 검토, 검증(verification) 개발작업이 프로젝트를 위해 선택된 프로세스와 방법에 맞게 수행되었는지 체크 / **Are we building the product RIGHT?**
 - 요구된 소프트웨어 결과물이 품질 수준에 맞게 생산되었는지 검사

- 확인(validation) : 개발 프로세스에 의하여 생성된 결과물의 (사용자 관점) 정확성을 체크 / **Are we building the product RIGHT?**
 - 정적 확인(static validation) : 소프트웨어를 실행시키지 않고 결과물의 정확성을 체크
 - 동적 확인(dynamic validation) : 소프트웨어를 실행시켜 잘 작동하는지 확인

- 테스트

- 동적 확인 작업, 테스트 결과가 예상되는 결과와 일치하는지 체크

- Walkthrough & Inspection

프로젝트 관리

- 프로세스와 관련된 이슈를 적절히 관리
 - 작업 과정에 **자원**을 어떤 작업에 할당할 것인지 기술
 - 소프트웨어 **계획**은 프로젝트의 개발 프로세스를 모니터링하고 제어하는데 사용되는 기준이 됨

● 관리 메트릭

● 객관적인 데이터가 필요. 소프트웨어 메트릭이 사용

- 프로덕트 메트릭 : 개발한 프로덕트, 소프트웨어 자체의 특성을 계량화

ex) fan-in/fan-out, length of code, fog index (대학생은

13~16)

- 프로세스 메트릭 : 소프트웨어 개발에 사용된 프로세스의 생산성을 계량

ex) Cost of poor quality (저품질로 발생하는 비용) =

$\text{rework effort} / \text{total effort} \times 100.$

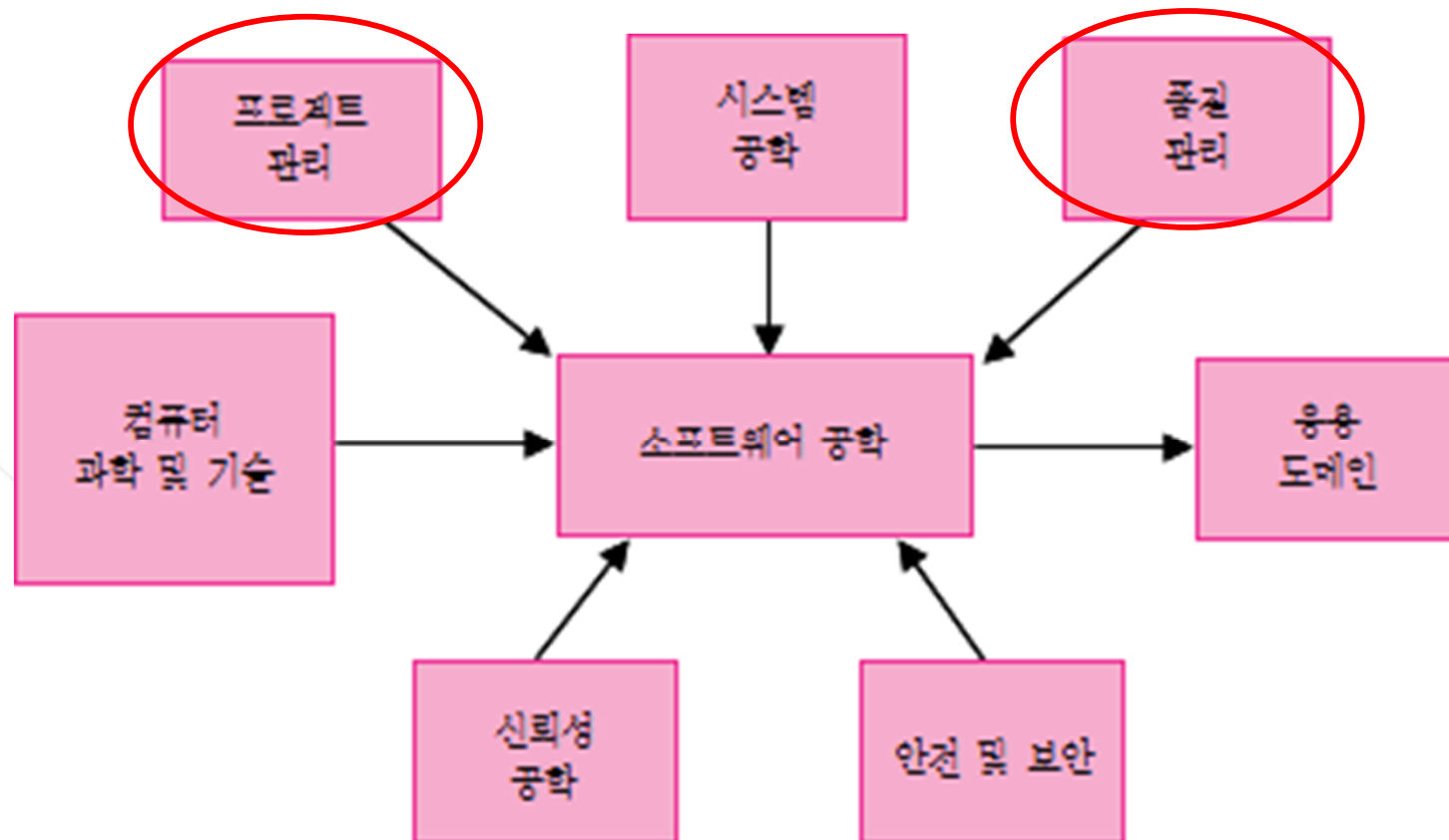
- 참고: 프로젝트 메트릭:

ex) Schedule Variance,
Effort Variance,

$$RSI(\text{Req. Stability Index}) = (1 - (\text{변경, 추가, 삭제 요구 개수} / \text{초기 요구 개수})) \times 100.$$

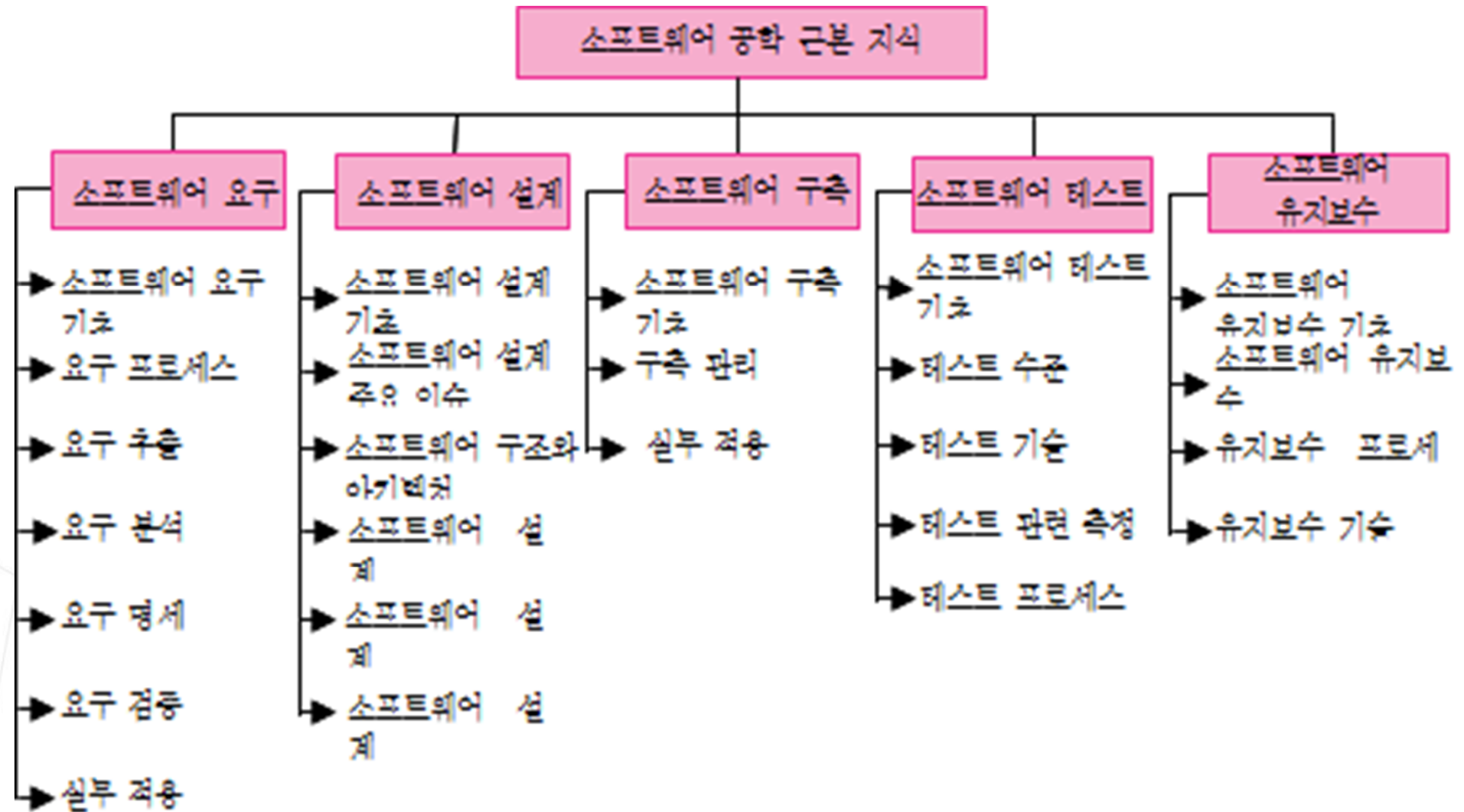
1.5 소프트웨어 공학 지식 체계

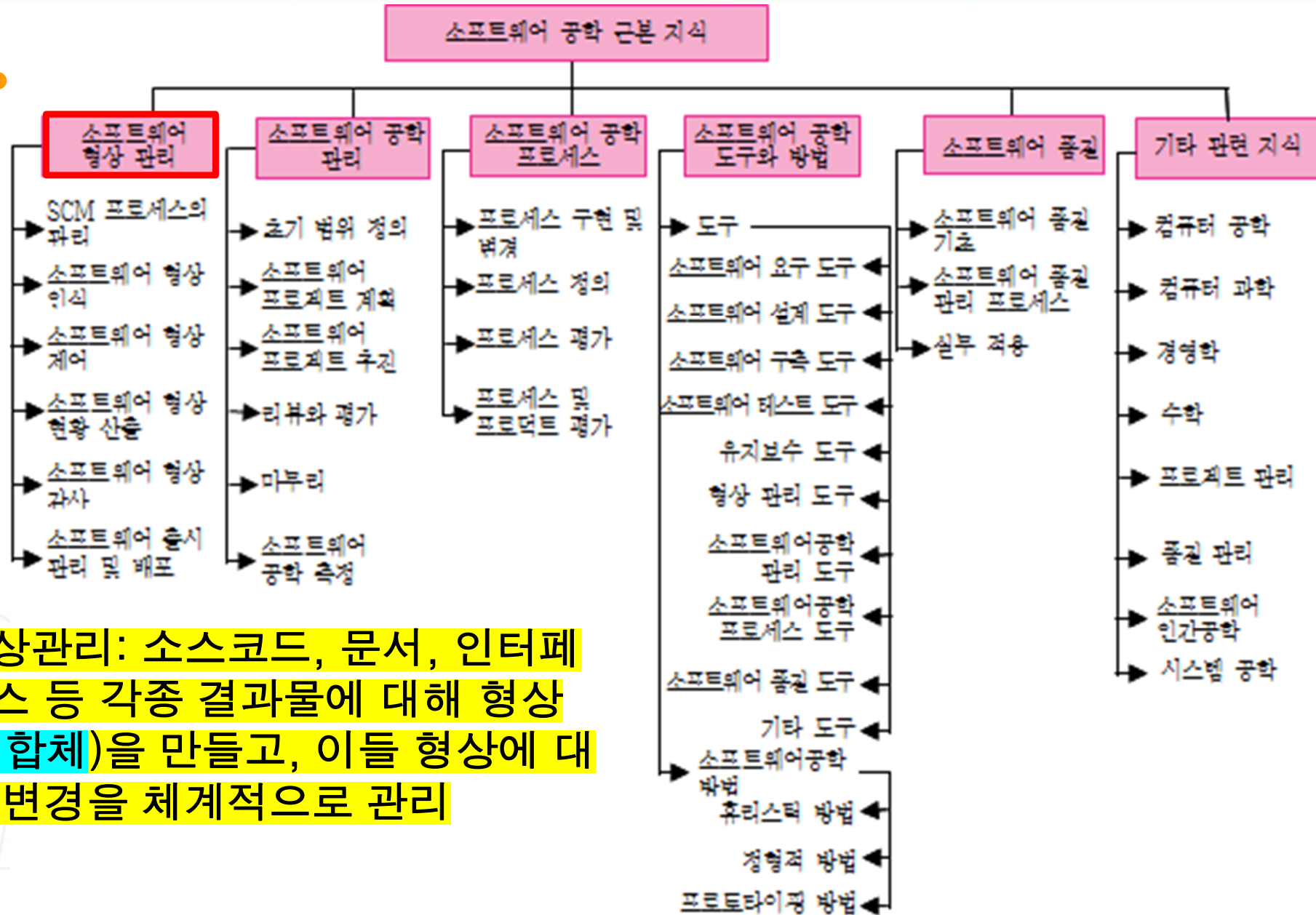
- 다른 분야와의 관계:



SWEBOK (Software Eng. Body of Knowledge)

IEEE & ACM (2004)





형상관리: 소스코드, 문서, 인터페이스 등 각종 결과물에 대해 형상 (집합체)을 만들고, 이들 형상에 대한 변경을 체계적으로 관리



새로 쓴 소프트웨어 공학

New Software Engineering