



소프트웨어 인스펙션 효과

Soft  Soft
www.soft4soft.com



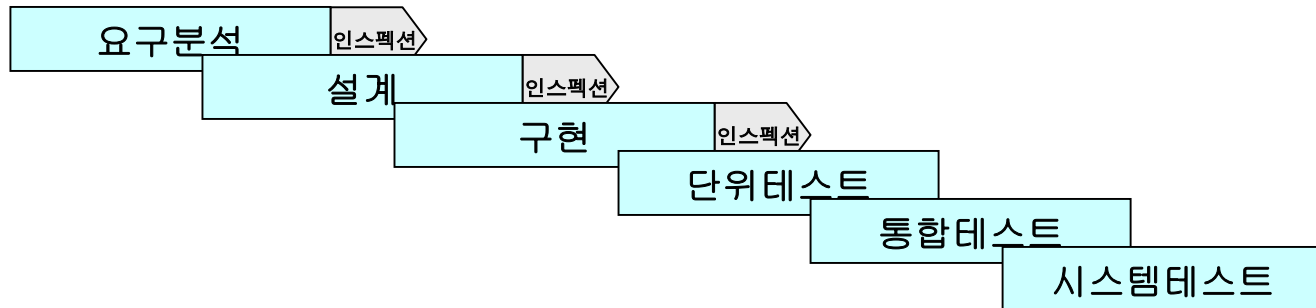
Contents

- 소프트웨어 및 코드 인스펙션 정의
- 소프트웨어 결함 및 적용 기술
- 생명주기상 결함 효과
- 코드 인스펙션
- CMM Levels 정의
- CMM Level 4와 SQA
- CMM Levels과 코드 결함 수
- 코드 인스펙션 도입 사례
- 코드 인스펙션 도입 효과
- SW 품질 관리 및 검증 효과

소프트웨어 및 코드 인스펙션 정의

- 소프트웨어 인스펙션 정의
 - 소프트웨어의 결함을 발견하기 위하여 개발 과정상의 결과물을 검토하는 것

- 소프트웨어 인스펙션 대상
 - 요구사항, 설계 등의 문서 산출물
 - 개발 단계의 소스코드



- 코드 인스펙션
 - 개발 단계의 소스코드를 인스펙션하여 해당 소스코드에 내재되어 있는 직.간접적인 결함, 성능 및 메모리의 문제점을 검출

소프트웨어 결함 및 적용 기술

- 소프트웨어 결함 분포 J.E.Heiser, An Overview of Software Testing. IEEE TOSE 1997

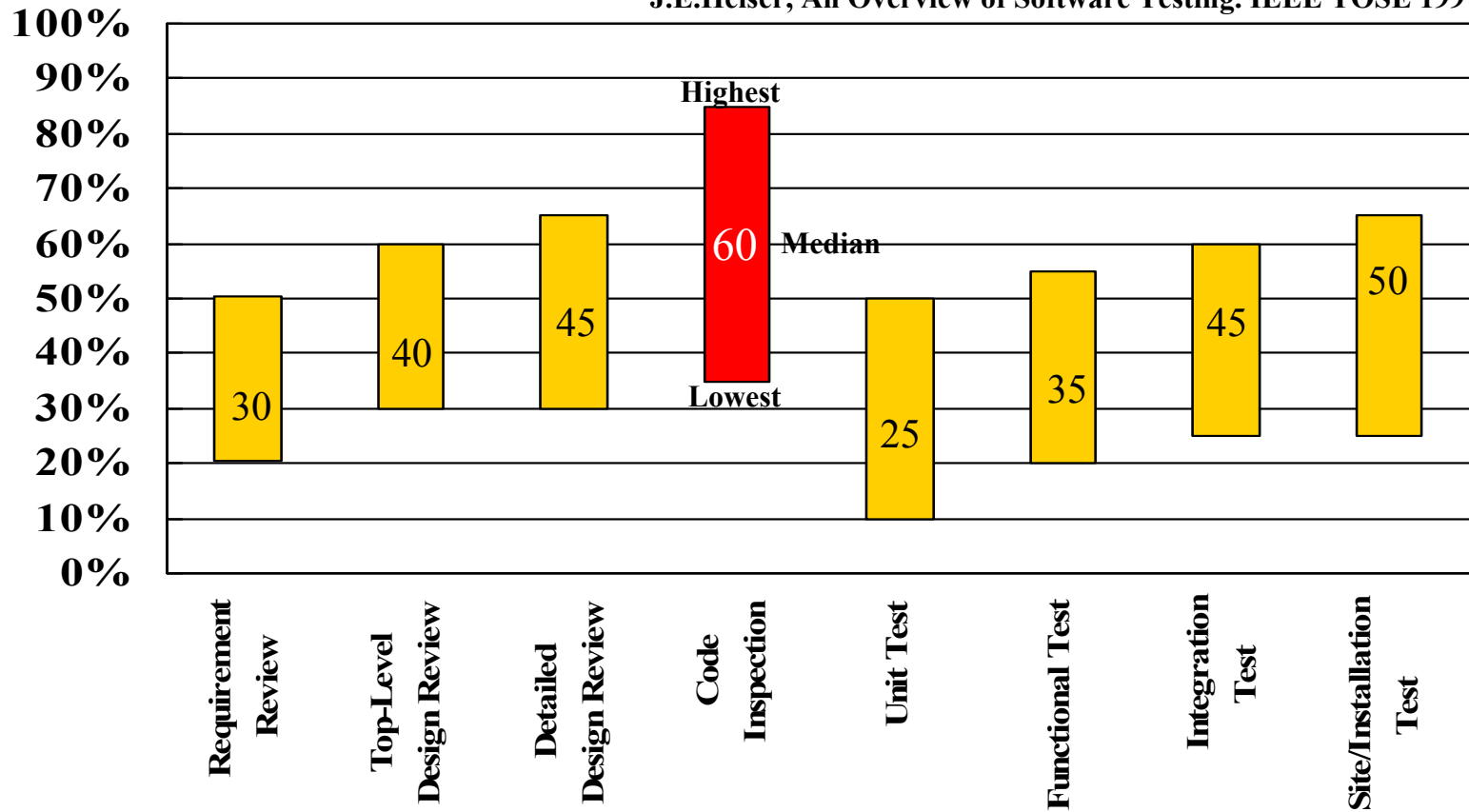
SW Lifecycle	Defect Ratios	Defects per FP
Requirements	20%	1.00
Design	30%	1.25
Coding	35%	1.75
Documentation	5%	0.60
Bad Fixes	10%	0.40
Total	100%	5.00

- 소프트웨어 결함 제거 기술의 효과

Life Cycle Techniques	Req. Defect	Design Defect	Code Defect	Documentation Defect	Performance Defect
Review, Inspection	Fair	Exc.	Exc.	Good	Fair
Prototype	Good	Fair	Fair	N/A	Good
Dynamic Test	Poor	Poor	Good	Fair	Exc.
Program Proof	Poor	Poor	Good	Fair	Poor

생명주기상 결함 제거 효과

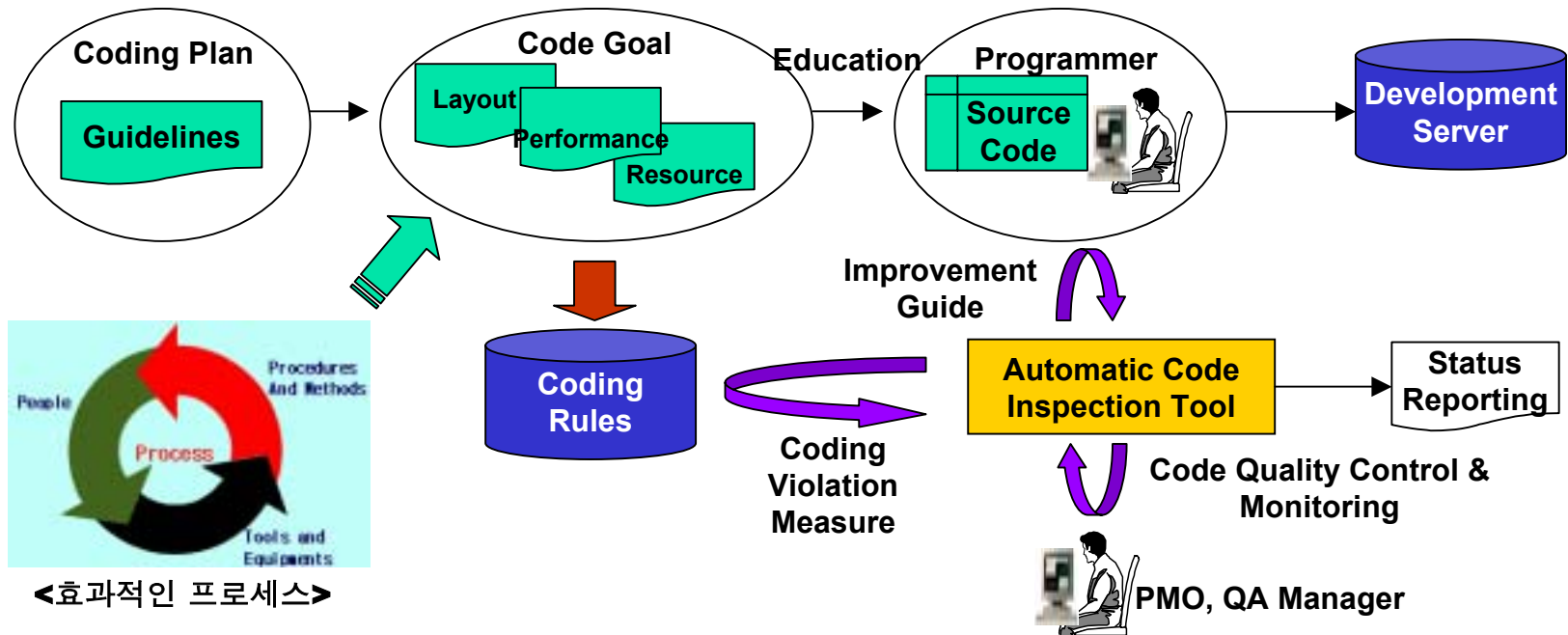
J.E.Heiser, An Overview of Software Testing, IEEE TOSE 1997



코드 인스펙션

■ 코드 검사 점검 활동 - CMM 4 수준

- **Quality Plan** : 코드 표준 정의서
- **Quality Goal** : Zero-Defect 코드 품질 목표
- **Quality Measure** : 코드 품질 목표를 달성하기 위한 품질 평가
- **Quality Control** : 코드 품질을 정량적으로 관리하여 품질 목표를 달성



코드 인스펙션

■ 코드 인스펙션과 결함 유형

Inspection Type	Defect Type	Inspection Check
Coding Standard Violation	Coding Style	일관된 코딩 구조와 스타일 유지 코드의 가독성 및 이해성 향상 코드의 유지 관리성 향상 및 비용 절감 (효과) 규격화된 우수한 코드 작성
	Naming Convention	
	Hard Coding	
Potential error or Actual error	Data errors	SW 품질 향상 - 성능 향상 - 메모리 누수 예방 (효과) 고급 코딩기술과 프로그래밍 기능 사용 유도
	Control errors	
	Input/output errors	
	Interface errors	
	Exception management errors	
	Storage management errors	

Powerful
Code
Inspection
Rules

코드 인스펙션 : MISRA-C



- MISRA-C (Motor Industry Software Reliability Association)
 - 자동차(우주 항공, 통신, 의료, 국방 등) 제조 분야 C 코딩 가이드 라인 제시
 - ISO/IEC C 표준 9899:1990/1999 기반으로 한 표준 문서
- MISRA-C 가이드라인(www.misra.org.uk)
 - 안전성과 관련된 고급언어에서 프로그램 되어야 하는 코딩을 위하여 Safer C 지침을 개발 한 회사들의 모임
 - 코딩 시 적용 가능한 141개의 코딩 규칙
-121 mandatory, 20 advisory
- MISRA 운영 위원회
 - AB Automotive Electronics Ltd
 - Ford Motor Company Ltd
 - Jaguar Cars Ltd
 - Lotus Engineering
 - MIRA Ltd
 - Ricardo Tarragon Ltd
 - TRW Automotive Electronics

CMM Levels 정의

- CMM Level 3 (정의됨)
 - 제품과 프로세스가 표준화되고 잘 정의되는 단계(관리/통제되지 않은 단계)
- CMM Level 4 (관리됨)
 - 제품과 프로세스가 관리되고 통제되는 단계
 - 제품과 프로세스를 정량적으로 측정/평가되어 그 결과를 관리하는 단계
 - 수행 결과에 대한 데이터를 수집하고 분석하여 결과를 피드백하는 단계

Level 5: Optimizing (계속적인 프로세스 개선)

Level 4: Advanced (소프트웨어 제품과 프로세스에 대한 자세한 측정치)

Key: SW Quality/Quantity Process Management

Level 3: Defined (효과적인 소프트웨어 공학 기술 적용)

Level 2: Repeatable (기본적인 프로젝트 관리 프로세스)

Level 1: Initial (가장 기초적인 개발 환경)



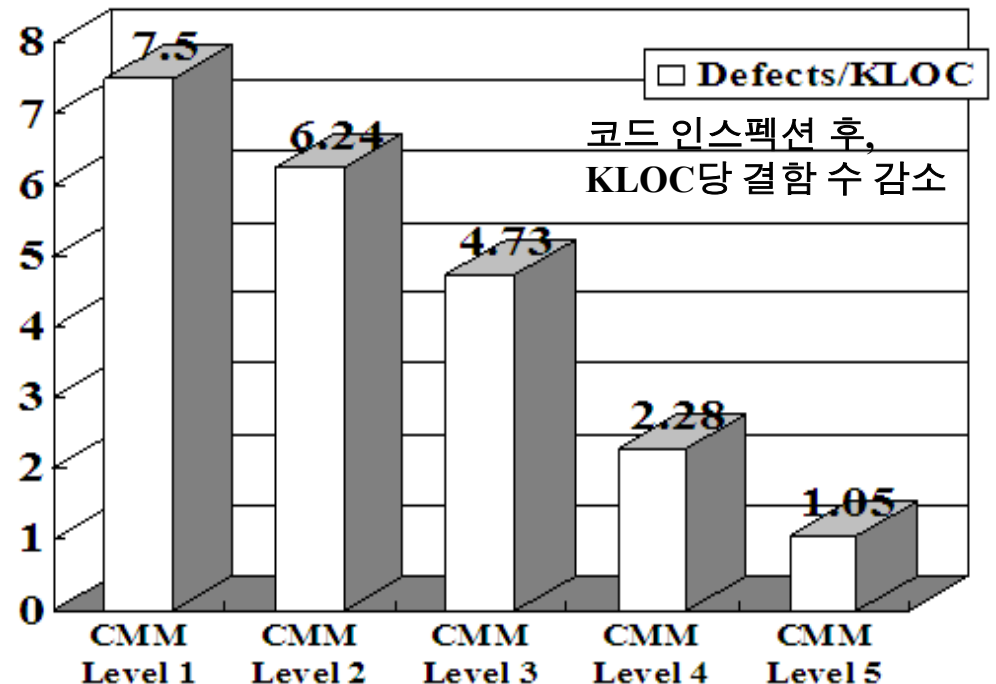
CMM Level 4와 SQA

- **Focus**
 - **Product and process quality**
- **Key Process Areas**
 - **Quantitative process management**
 - **Software quality management**
- **Software quality management Goal**
 - 프로젝트의 SW 품질 관리 활동을 계획한다
 - SW 산출물 품질의 평가 목표와 그 우선 순위를 정의한다
 - SW 산출물의 품질 목표를 달성하기 위한 실제 진척을 평가하고 관리한다
- **SQA (Software Quality Assurance)의 정량적인 품질관리**
 - **Quality Plan** : SW 및 코드 표준 정의서
 - **Quality Goal** : 정량적인 품질의 목표를 설정
 - **Quality Measure** : SW 품질 목표를 달성하기 위한 품질 평가
 - **Quality Control** : SW 산출물에 대한 품질을 정량적으로 관리하여 품질 목표를 달성
- **SQA를 위한 RESORT의 품질관리 도구**
 - **SW Quality Tool** – SW 품질
 - **Code Checker Tool** – 코드 품질

CMM Levels과 코드 결함 수

출처: CMU SEI TR, Sept. 2003-014

- Level 5 (최적화) Best -Quality Controlled and Repeatable Results
- Level 4 (정량적 관리/통제됨) Operating with Checks and Balances
- Level 3 (표준화/정의됨) Coding Standards must be Enforced
- Level 2 (반복) An Organization must have a Coding Standards
- Level 1 (초기) Results – Not Controlled and Non Repeatable



□ Industrial Average:
• 6 - 30 Defects/KLOC

□ 휴대폰 SW(약 500만 라인)
• 3,000 Defects

□ 우주왕복선 SW(약 500만 라인)
• 500 Defects

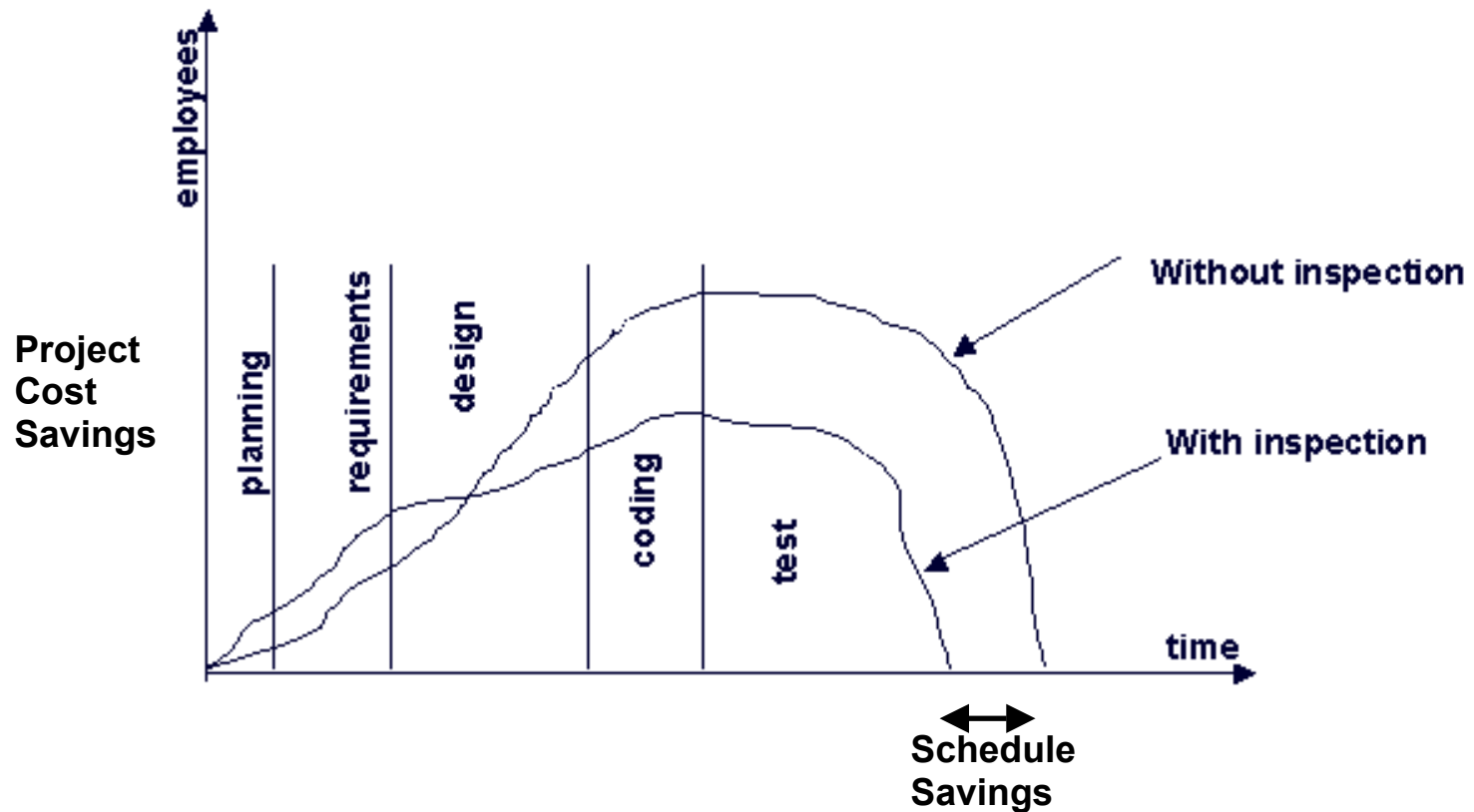
코드 인스펙션 도입 사례

■ 인스펙션도입 사례

회사	도입 효과
AT&T	텔레커뮤니케이션 시스템 제품에 대해서 요구, 디자인, 코드, 테스트 계획 부분에 대해서 인스펙션 한 결과 인스펙션은 생산성에서 14% 의 증가를 보였으며, 품질에서 10배 의 증가를 보였고 테스트보다 20배 이상의 효율을 보였음이 보고되었다
HP	어떤 제품에 대해서 코드 인스펙션을 사용한 결과 시간당 0.2 개의 오류를 발견하였으며, 다른 테스트에 의한 것보다 80% 의 다른 오류를 발견하였음이 보고되었다
Bell-Northern Research	텔레커뮤니케이션 시스템 제품에 대하여 코드 인스펙션을 적용한 결과 Man-hour 당 1 개의 오류를 발견하였으며 테스트보다 인스펙션은 20배 이상 효율적이라는 사실이 보고되었다
IBM	운영체제 제품을 개발하면서 디자인과 코드 인스펙션을 적용한 결과 코딩 생산성이 23% 증가하였으며, 단위 테스트 후 오류 검출에서 38% 감소하였으며 오류 검출 당 1.1hour 가 소요됨이 보고되었다

코드 인스펙션 도입 효과

- 인스펙션 도입 전.후 효과 (출처: **Michael Fagan**)



코드 인스펙션 도입 효과

■ 인스펙션도입 효과

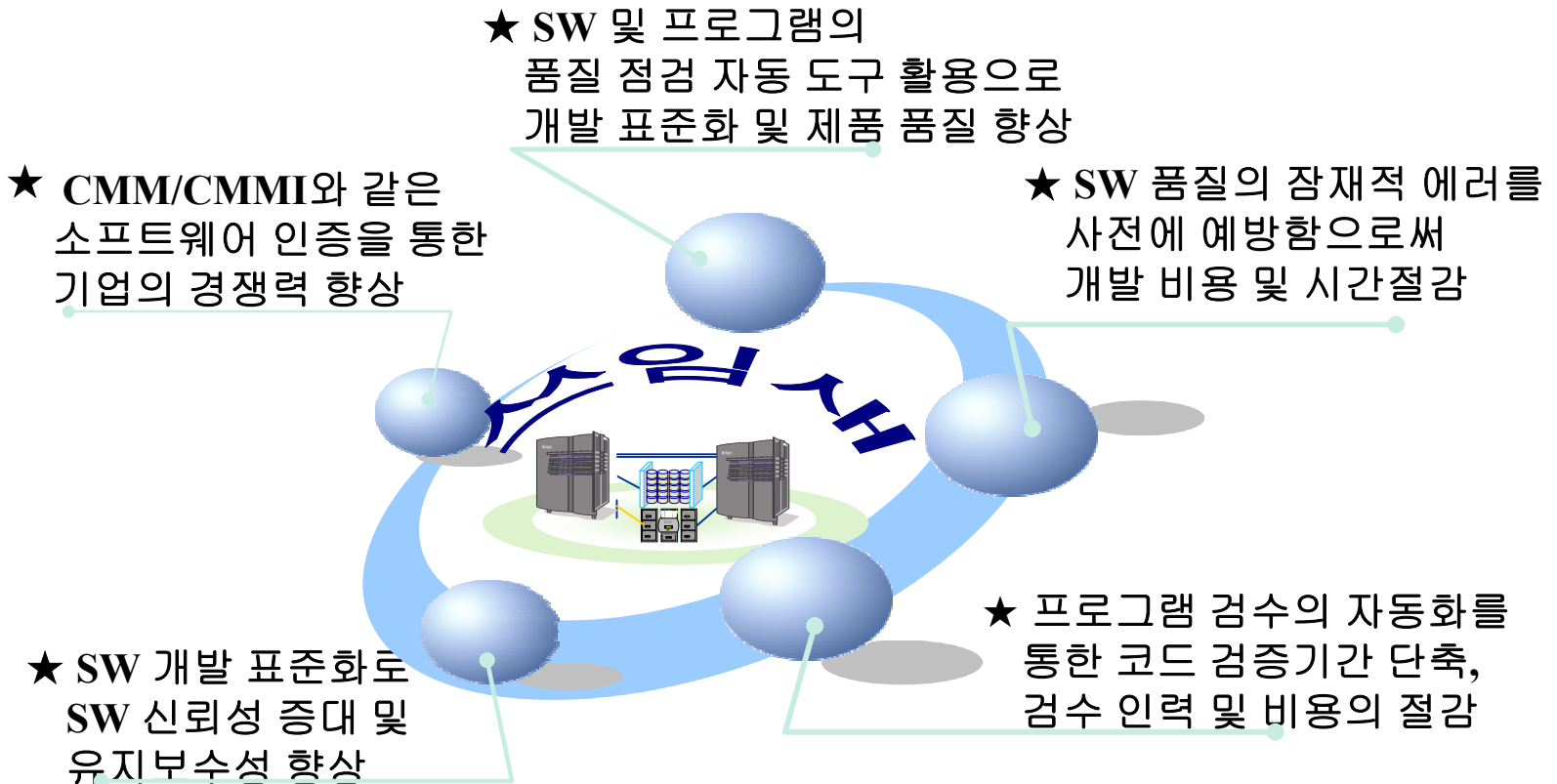
Benefits	Inspection versus Testing [IBM Journal, HP Journal]
에러 예방	개발 초기에 에러 예방 테스팅 단계로 전파되는 에러를 적어도 60-80% 정도 줄임 (82%(IBM), 80%(HP) 제거)
비용 절감	인스펙션을 한 시간 동안 효과적으로 수행하는 것은 8~12시간 정도의 테스트 작업 절약 테스팅시 에러 발견 및 수정에 드는 비용 60-80% 정도 절약
도입 효과	프로젝트 팀간의 대화 향상 개발 및 테스트 프로세스의 품질 및 생산성 향상 비용 및 개발 기간 단축 유지보수 노력 절감
중요성	"하나의 좋은 인스펙션은 30,000 개의 테스트 케이스와 동등한 효과를 가져올 수 있다." - Vern Crandell

SW 품질 관리 및 검증 효과

Increases
-Quality
-Productivity
-Profitability



Reduces
-Cost
-Time
-Risk





If you cannot MEASURE it, you cannot IMPROVE it

(주)소프트4소프트

대전광역시 유성구 문지동 103-6 ICU 창업보육센터 T215호

Tel : 042-866-6632~3

Fax: 042-866-6626

구매 및 데모 문의 : sales@soft4soft.com

기술 및 일반 문의 : info@soft4soft.com

www.soft4soft.com