

- 위험 관리의 목적은 위험이 발생되었을 때의 영향을 줄이는 것이다.
- 모든 프로젝트는 시간(일정), 비용(예산), 범위(품질, 기능) 제약을 가짐 → 이들을 지키지 못할 위험 발생 할 수 있음
- Boehm이 제안한 위험 관리[Boehm, 1991]
 - 위험파악 (source: 기술적, 관리적(시간, 비용, 범위), social 등)
 - 위험 분석 (발생 확률, 위험정도 등)
 - 위험 우선순위 정하기
 - 위험 해결
 - 위험 모니터링

위험 파악(1) – Boehm의 10대 위험

위험 요소

1.인력 부족

2.비현실적 일정 및 예산

3.잘못된 기능의 소프트웨어 개발

위험 관리 기법

- 유능한 인력모집, 팀 구성, 요원 배치, 교차교육, 유능 인력 사전 확보
- 더 자세한 비용, 일정 예측, 원가 분석, 점증적 개발, 소프트웨어 재사용 요구를 줄임
- 사용자 회람, 프로토타이핑, 사용자 지침서를 조기에 작성, 조직 분석, 직능 분석

위험 파악(1) – Boehm의 10대 위험

위험 요소

4. 잘못된 인터페이스의 개발

5. 과포장 (필요
지 않을 좋은
기능을 추가)

Gold plating

6. 계속적인
요구 변경

위험 관리 기법

- 프로토타이핑, 시나리오, 태스크 분석,
사용자 분류(기능, 스타일, 업무)

- 요구 삭감, 프로토타이핑,
비용-수익 분석, 원가 분석

⇒ lean 개발방법.

- 최대 변경 상한선, 정보 은닉,
점증적 개발(다음 버전까지 변경을
연기)

위험 파악(2)

위험 요소

위험 관리 기법

- 7. 외부 (outsourced) - 벤치마킹; 검사; 대조 확인;
모양의 빈약 성숙도 분석
- 8. 외부(outsourced) - 대조 확인; 사전 검증; 설계 경연;
기능의 빈약 팀 작업
- 9. 실시간 성능
의 빈약 - 시뮬레이션, 벤치마킹, 모델링,
프로토타이핑, 튜닝
- 10. 기술적 취약 - 기술 분석, 비용-수익 분석,
프로토타이핑; 점검

위험 파악(2)

불필요한 detail을 외부에 숨김. 이를 위한 대표적인 방법이 처리(operation)을 interface와 implementation(detail)으로 나누고, 외부에는 인터페이스만 공개함.

In object-oriented programming, Information hiding reduces software development risk by shifting the code's dependency on an uncertain implementation onto a well-defined interface. Clients of the interface perform operations purely through it so if the implementation changes, the clients do not have to change.

위험 분석과 우선순위 정하기

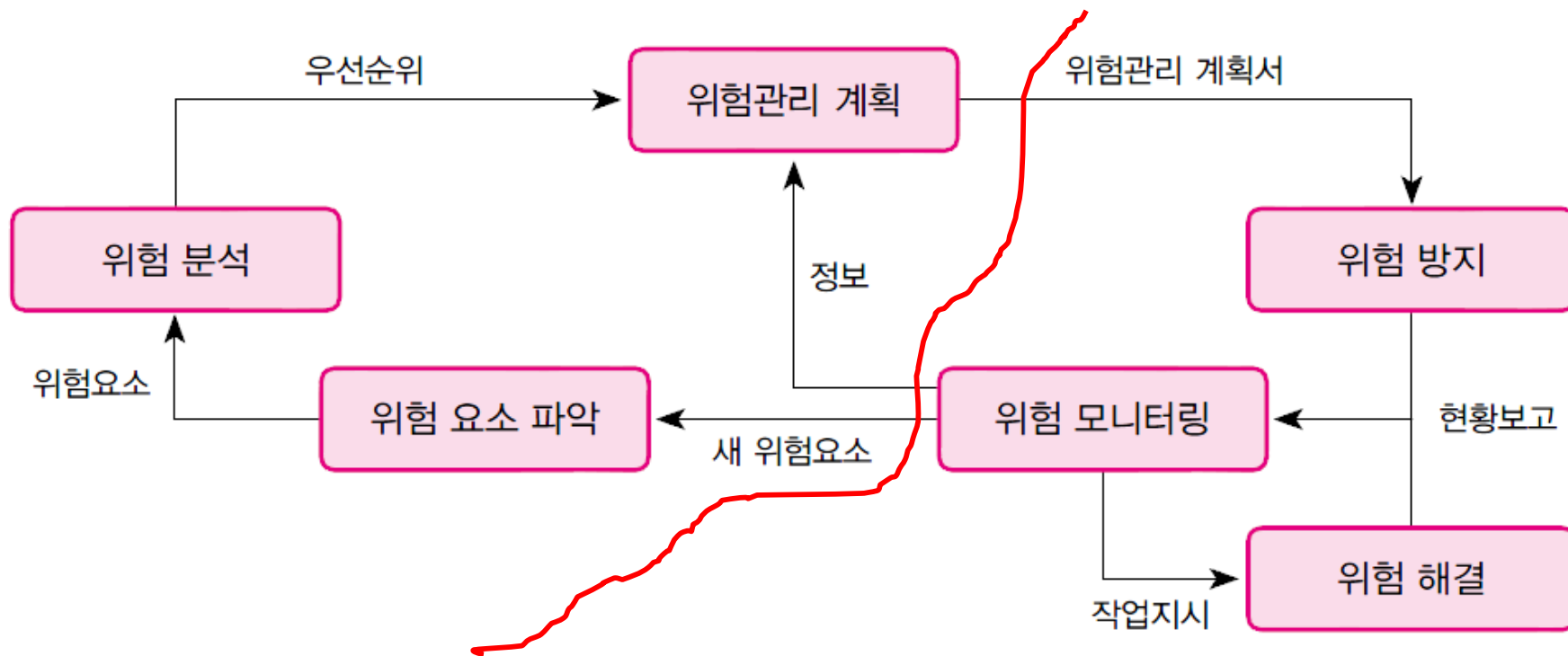
- 위험 분석은 각 위험에 대한 피해 정도, 위험 해결 방법, 이에 대한 비용들을 결정하는 것이다.
 - 손실 발생 확률
 - 손실 발생 규모
 - 위험 노출(exposure) : 발생확률 * 손실규모

Example:

Risk	Probability	Loss	Exposure
Overall response time too slow	0.20	\$4,000.00	\$800.00
Web server fails	0.25	\$5,000.00	\$1,250.00
User IDs can be viewed	0.10	\$50,000.00	\$5,000.00

위험 해결과 모니터링

- 위험 해결은 위험 관리 계획에 명시된 위험을 줄이는 기법을 구현하고 실행하는 것.



위험 관리 프로세스

3.6 계획서 작성

1 개 요

1.1 프로젝트 개요

1.2 프로젝트의 산출물

1.3 정의, 약어

2 자원 및 일정 예측

2.1 자원

가. 인력

나. 비용

2.2 일정

3 조직 구성 및 인력 배치

3.1 조직 구성

3.2 직무 기술

계획서 작성

4 WBS

5 기술관리 방법

5.1 변경 관리

5.2 위험 관리

5.3 비용 및 진도 관리

5.4 문제점 해결 방안

6 표준 및 개발 절차

6.1 개발 방법론

7 검토 회의

7.1 검토회 일정

7.2 검토회 진행 방법

7.3 검토회 후속 조치

8 개발 환경

9 성능 시험 방법

10 문서화

11 유지보수

12 설치, 인수

13 참고문헌 및 부록

