제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ☆ 📖 솔루션:💡 📘📒✔️

😀 확실성과 보안성

배경

- 현대인의 삶이 컴퓨터 시스템에 매우 의존적이다.
- 의존도가 높을 수록 장애로 인해 피해도 커진다
    - 금전적 손실, 명성추락, 환경적 손실, 인명손실
- 피해를 줄이기 위해서는
    - 약속된 기능을 제공해야 한다.(1장 목표)
    - 확실해야 한다.(dependable) (2장 목표)
    - 관리를 잘해야 한다. (3장 목표)
    - 등등 수없이 많다.

복 신 안 가 보안성	
RAMS (시스템 분야)	**Dependability (SW)(★ 책 292.p)**
Reliability	**가용성**
Availability	**신뢰성**
Maintainability	**안전성**
Safety	**보안성**
	복원성

😎확실성있는 시스템

📜시스템 확실성의 개념

- **가장 중요한 속성**
    - dependability
- **사용자의 신뢰 정도**
    - The User;s degree of Trust
    - The User’s Confidence
- **주요 확실성 속성**
    - 가용성
    - 신뢰성
    - 안전성
    - 보안성
    - 복원성

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜확실성의 중요성 (⭐ 책 292.p)

- **장애는 많은 사람에게 영향을 미침**
  - widespread effects
  - 많은 시스템들이 거의 사용되지 않는 기능을 포함하고 있다.

- **확실성이 없으면 ,사용자가 시스템을 거부함**
  - rejected
  - 시스템이 신뢰성이나 보안성이 없다는 것을 사용자가 발견한다면 사용하기를 거부할 것이다.

- **시스템 장애 비용이 막대함**
  - 실패가 경제적 손실을 이끌거나 물리적 피해를 이끈다면 실패에 대한 비용손실은 아마도 클것이다.

- **확실성이없는 시스템은 정보손실을 초래할 수 있다.**
  - 데이터를 수집하고 유지하는데 매우 많은 비용이 든다.

📜장애의 원인 (⭐ 책 293.p)

- **하드웨어 장애**
  - 랜덤하게 발생
  - **항상 하드웨어가 정상 동작하는지 확인**이 필요하다

- **소프트웨어 장애**
  - **명세화, 설계, 혹은 구현시의 실수 때문**에 시스템 소프트웨어 장애가 발생한다.
  - 설계는 첫 단추가 중요하다

- **운영시의 장애**
  - **설계자가 의도한 대로 시스템을 사용하지 않거나 운영하지 않아서 장애가 발생**
  - 사람 실수로 발생
  - 사람이 SW를 쓰기 때문에 실수가 발생할 수 있다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📙✔️

✔️중대한 시스템이란? (critical system) -> 확실성 있게 개발해야한다

시스템 장애가 사람의 부상, 환경훼손, 광범위한 경제적 손실을 초래할 수 있는 시스템의 종류를 중대한 시스템이라 한다.

- 시스템 장애가 치명적 손실을 초래한다.
- 장애가 매우 드물게 발생하도록 개발되어야 한다
- 장애 발생시 복구 매커니즘을 포함해야 한다.

📣 | Zero-defect : 무결점 시스템

✔️📌중대한 시스템의 예

- **Safety critical system**
  - 예시)
    - 원자력 발전제어 시스템
    - 자율주행 시스템
  - **장애로인해서 사람에게 큰 위험 가능성**
- **Mission critical system**
  - 장애가 발생하면 **중대한 임무를 수행하지 못함**
- **Business critical system**
  - 장애가 발생하면 **업무 비지니스에 막대한 영향**

📣 **| 장애가 발생하면 빠르게 복구가 가능해야한다.**

📌📣 **|오늘은 5가지 특성을 확실히 알아가는것이 중요**

📌**Dependability (SW)(⭐ 책 292.p)**		
가용성 Availability	**수치적**으로 나타낸다.(%)	📘 시스템이 서비스를 요청받았을 때 해당 서비스를 제공할 수 있는 능력
신뢰성 Reliability	서비스를 제공할 확률	📘 시스템이 명세화된 대로 서비스를 제공할 수 있는 능력
안전성 Safety	📣 **내부적**인것(내부의 침입) 에 대해서 단속 📘 치명적인 장애로 발전하지않게 대체할 수 있는 능력	
보안성 Security	📣 **외부**의 침입에대해서 시스템을 보호할 수 있는 능력 📘 시스템이 의도적 혹은 우연적 침입을 막아낼 수 있는 능력	
복원성 Resilience	📣 장애가 발생되었을 때 시스템을 원래 상태로 복구시킬 수 있는 능력 📘 시스템이 손상 사건에 견디고 복원할 수 있는 능력	

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📣 교수님 추가 말씀

SIL 이란? → 안전 무결성 기준

인증기관에 system의 safty case를 제출하면 sil 등급을 매겨 안전 인증서를 발급한다

안전 무결성 수준은 안전 기능에 의해 제공되는 상대적 위험 감소 수준으로 정의되거나 목표 위험 감소 수준
간단히 말해 SIL은 안전 계측 기능에 필요한 성능을 측정하는 것

Safety Integrity Level	Safety	Probability of Failure on Demand	Risk Reduction Factor
SIL 4	> 99.99%	0.001% to 0.01%	100,000 to 10,000
SIL 3	99.9% to 99.99%	0.01% to 0.1%	10,000 to 1,000
SIL 2	99% to 99.9%	0.1% to 1%	1,000 to 100
SIL 1	90% to 99%	1% to 10%	100 to 10

CC 인증
https://m.blog.naver.com/kebinj/40102557431
CC(Common Criteria) 인증은 IT 제품(하드웨어, 펌웨어, 소프트웨어)의 보안 기능성과 평가 과정에서 그 제품들에 적용되는 보증수단에 대한 공통의 요구 사항들을 제시함으로써, 독립적으로 수행된 보안성 평가의 결과들을 비교할 수 있도록 한다.
평가결과는 소비자가 IT 제품이 그들의 보안 요구를 충족시키는지 결정하는 데 도움을 줄 수 있다.

- 보안성에서 사용
- Common Criteria(공통기준)
- 1~7단계가 있음

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️속성간의 관계(💡책에없는 내용)

- **안전성**은 **가용성**과 **신뢰성**에 의존
- **외부 공격**을 당하면 시스템을 **신뢰**할 수 없음
- **DOS공격**은 **가용성을 손상**하는 것이 목적 임
- **바이러스에 감염**되면, **신뢰성**과 **안전성**을 확신할 수 없음

📜확실성 달성을 위한 점검 사항(⭐296.p)

- **SW 명세화 및 개발과정에서 우발적 오류를 피한다**
- **V&V 프로세스를 갖춘다**
  - **확실성**에 영향을 주는 잔여 오류를 발견하는데 특효
- **결함 내성을 갖도록 설계한다 (건물의 내진 설계와 유사)**
  - 장애가 발생한 후에도 작업이 가능
- **보호 메커니즘을 설계한다**
  - **가용성 및 보안성**을 손상시킬 수 있는 외부 공격으로부터 보호하는 메커니즘을 설계
- **설정을 정확히 한다**
  - 운영환경에 맞게 시스템 및 지원 소프트웨어를 구성한다.
- **공격에 저항해야 한다.**
- **신속히 복구되도록 설계한다.**

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜결함내성 (⭐297.p)

> 확실성 있는 시스템이 자신을 모니터링하고, 잘못된 상태를 감지하고, 장애가 발생하기 전에 결함으로부터 복구하기 위한 여분의 코드를 포함해야 한다는것을 의미

- MooN (M out of N)
    - 1oo2 (1 out of 2)
    - 1oo3 (1 out of 3)
    - 2oo3 (2 out of 3)

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜시스템과 소프트웨어

- 소프트웨어 공학은 고립된 활동이 아니다.
    - 📣 **| 만들고자 하는 SW의 주변 환경에 대한 이해가 필요하다.**
- 고립된 시스템이 아니라 인간적, 사회적,조직적, 목적을 갖는 더 광범위한 시스템의 일부이다.
    - 사회의 일부가 된다.

✔️📌사회 기술적 시스템 스택 (⭐299.p)

📣 | 기말 시험에 냈던거야~

- 광범위한 시스템을 사회기술적 시스템이라고 함.

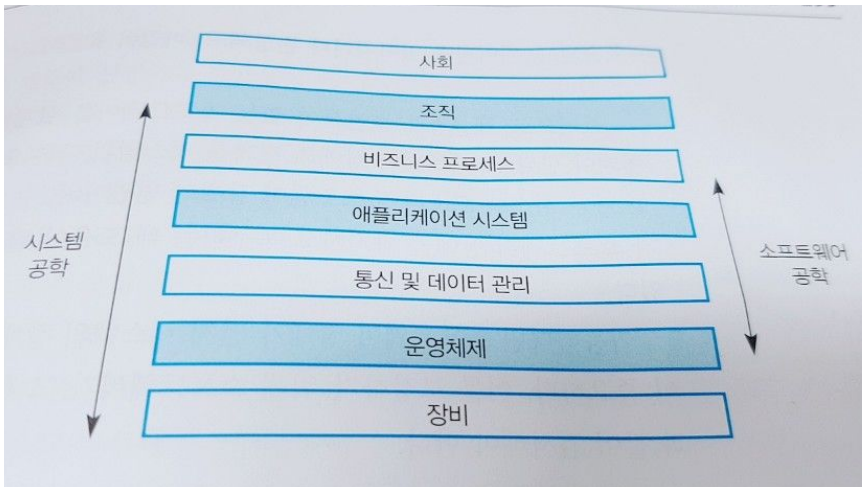


그림 | 📖 책 299p. **사조 비애 통운장**

📖 책 298p.
사회기술적 시스템은 컴퓨터, 소프트웨어,기타 장비와 같은 기술적 구성 요소뿐만 아니라 사람, 프로세스, 규정과 같은 비기술적 요소를 포함한다.

- 가능한 한 소프트웨어 장애가 그 장애가 발생한 계층에만 국한되고 시스템의 다른 계층의 작동에 심각한 영향을 미치지 않는다.

- 시스템의 다른 계층의 결함과 장애가 해당 소프트웨어에 어떻게 영향을 줄 수 있는지를 이해한다. 또한 이러한 장애를 감지하기 위한 검사가 소프트웨어에 내장될 수 있는 방법을 고려하고, 장애 복구를 위한 지원이 어떻게 제공될 수 있는지를 고려한다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜규정과 준수 (Regulation and Compliance)

📖 301.p | 안전성 중심 시스템을 개발하는 기업은 인증을 받기 위해 규칙과 규정을 준수한다.

✔️규제 받는 시스템

- 대부분의 중대한 시스템은 **규제 시스템**이며 **승인** 받아야함
    - 예시)
        - 원자력 시스템
        - 항공제어 시스템
        - 의료 장비

✔️안전성 규제

- **규제와 준수는 사회기술적 시스템 전체에 대해서 적용됨**
- 안전 관련된 시스템 **인증**을 받아야 함
- 인증서를 얻기 위해서, **Safety case**를 작성
    - (📖 301.p)
      📣 시스템이 인증기관에 인증요청을 할때 제출하는 것이 **Safety case**이다.
        - 책에서는 이것을 **안전성 사례**라고 소개가 되어 있다.
        - 📣 안정성사례 → 종합 안전 대책 보고서

- **인증을 위한 문서화는 비용이 크다.**
    - 인증을 위한 문서를 개발하는 것이 시스템 자체를 개발하는 것 만큼 비용이 많이 들 수 있다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜다양성과 중복성 (⭐301.p)

- **Redundancy 중복성**
  - 시스템의 일부가 실패할 경우에 사용될 수 있는 여분의 기능을 시스템에 포함하는것
  - 📣 중복 : 백업서버를 여러개 둘 수 있다.
- **Diversity 다양성**
  - 시스템의 중복된 컴포넌트들을 서로 다른 종류로 하여 동일한 방식으로 실패하지 않을 확률을 높이는것
  - 📣 다양성 : 외부 침입자에대해 방어하기 위해 다양한 운영체제를 사용

확실성을 달성하고 향상시키기 위한 전략은 중복성과 다양성 모두에 의존한다.

📖 책 305p |	확실성 있는 프로세스는 신뢰성을 달성하기 위해 **중복성**과 **다양성**을 사용 이 둘은 종종 **같은 목적**을 가진 다양한 활동들을 포함한다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜확실성이 있는 프로세스 (⭐304.p)

홧츠 험프리 曰 sw 개발 프로세스의 질이 sw 제품의 질을 좌우한다.

📖 책 304p
확실성 있는 소프트웨어 프로세스는 확실성 있는 소프트웨어를 생산하도록 설계된 소프트웨어 프로세스이다. 다양성과 중복성은 확실성이 있는 소프트웨어 개발 프로세스의 설계에도 사용가능하다. **좋은 소프트웨어 프로세스가 적은 오류를 포함하여 실행 시에 장애 발생이 적은 소프트웨어로 이어질 가능성이 높다.**

- **장애없는 소프트웨어를 예방**하는 것이 중요하다.

- 📖 304.p | **명시적으로 정의된 프로세스는 소프트웨어 생산 프로세스를 구동하는 데 사용되는 정의된 프로세스 모델을 가진다.**
    - 개발팀이 프로세스 모델에 정의된 대로 프로세스를 준수했다는 증명하는 데이터가 프로세스 중에 수집되어야 한다.

📖 책 304p
개발사 : 시스템 개발사는 일반적으로 프로세스 모델과 함께 그 프로세스를 준수했다는 증거를 규제기관에 제시한다. **규제기관** : 규제기관은 그 프로세스가 **모든 참여자에 의해 일관되게 사용되는 것**과 **다른 개발 프로젝트에도 사용될 수 있다**는 것을 확신할 수 있어야 한다. ⟶ **프로세스가 명시적으로 정의되고 반복 가능해야 한다는 것을 의미한다.**

- **반복가능한 소프트웨어를 사용해야한다.**
    - 📖 305.p |
      반복 가능한 프로세스는 개인의 해석과 판단에 의존하지 않는다
    - 📖 305.p |
      프로세스는 누가 개발에 참여하는지에 관계없이 다른 팀원들로 여러 프로젝트에 걸쳐 반복될 수 있다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ☆ 📖 솔루션:💡 📘📒✔️

📜확실성이 있는 프로세스 특성

견문표 다양성 감사!(김영환)
감다 견문표!(김성현)

- **감사가능**
  - 프로세스 참여자 이외의 사람도 프로세스 표준이 준수되고 있는지를 점검하고 프로세스 개선을 제안하기 위해 프로세스를 이해할 수 있어야 한다.
- **다양성**
  - 프로세스는 중복되고 다양한 검증 및 확인 활동을 포함해야한다.
- **문서화 가능**
  - 프로세스는 프로세스 활동들을 명시하는정의된 프로세스 모델을 가져야 하고 이들 활동 중에 문서를 생성해야 한다..
- **견고성**
  - 프로세스는 개별 프로세스 활동 실패로부터 복구될 수 있어야 한다.
- **표준화**
  - 소프트웨어 생성과 문서화를 다루는 소프트웨어 개발 표준의 포괄적인 집합이 있어야 한다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ☆ 📖 솔루션:💡 📘📒✔️

📜확실성이 있는 프로세스에 포함되는 활동들

요구사항 검토	**요구사항이 가능한 완전하고 일관성 있는지 검토**
요구사항 관리	**요구사항의 변경이 통제되는 것과 제안된 요구사항 변경의 영향이 그 영향을 받는 모든 개발자에 의해 이해되는 것을 보장**
정형명세	**소프트웨어의 수학적 모델을 생성하고 분석한다.** 10.5 절에서 정형 명세 장점을 설명한다. 아마도 이것의 가장 중요한 장점은 시스템 요구 사항의 매우 상세한 분석을 강제한다는 것이다.이 분석 자체는 요구 사항 검토에서 누락되었을 수있는 요구 사항 문제를 발견해 낼 가능성이 높다.
시스템 모델링	소프트웨어 설계가 일련의 그래픽 모델로 명시적으로 문서화 되고, 요구사항과 이들 모델 사이의 관계가 명시적으로 문서화 된다.
설계 및 프로그램 인스펙션	**시스템의 다른 표현들이 다른 사람들에 의해 검사되고 점검된다.**
정적 분석	**자동화된 검사**가 프로그램의 소스 코드에 대해 수행된다.
테스트 계획 수립 및 관리	시스템 테스트의 포괄적인 집합이 설계된다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

😎신뢰성 공학

📜신뢰성 VS 비용

- 신뢰성을 높일수록, 비용도 증가한다.
    - **적정선**을 찾는것이 중요하다.

📖 용어 316.p
브라이언-란델의 사고 모델 • Chain of events   ○ 결함 (fault) -> 오류(error) -> 고장(failure) → 사고 → 사망 **사람의 오류 혹은 실수** • 시스템 결함에 이르게 하는 사람의 행동. **시스템 결함** • 시스템 오류로 이어지는 소프트웨어 시스템의 특성 **시스템 오류** • 시스템 사용자가 **예상하지 못한 시스템 행동으로 이어지는 실행중의 잘못된 시스템 상태**. **시스템 장애** • 시스템이 **사용자가 기대하는 서비스를 제공하지 못할 때** 일어나는 사건.

시스템 결함이 반드시 시스템 오류를 초래하지 않고, 시스템 오류가 반드시 시스템 장애를 초래하지 않는 이유 📖 319.p
1. **프로그램의 모든 코드가 실행되는 것은 아니다.** 소프트웨어가 사용되는 방식 때문에 결함 (예를 들어, 변수의 초기화 실패)을 포함하는 코드가 한 번도 실행되지 않을 수있다. 2. **오류는 일시적이다.** 결함이있는 코드의 실행으로 인해 상태 변수가 잘못된 값을 가 질수있다. 그러나 그 변수가 접근되어 시스템 장애를 야기하기 전에 그 변수를 올바른 값으로 다시 설정하는 다른 시스템 입력이 처리 될 수있다. 잘못된 값은 실제 효과가 없다. 3. **시스템이 결함 감지 및 보호 메커니즘을 포함 할 수있다.** 이는 시스템 서비스가 영향을 받기 전에 잘못된 행동이 발견되고 정정되는 것을 보장한다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜📌Fault management (⭐317.p)

📣 | 시험에도 꼭나오는 부분이야~

- **고장을 줄이는 방법 (신뢰성을 향상 시키는 보완적 접근법)**

📖 책 317~318p [결함의 **회 감정 내**]	
결함회피	소프트웨어 설계 및 구현 프로세스는 **설계 및 프로그래밍 오류를 피하는 소프트웨어 개발 방식을 사용**하여 시스템에 도입되는 결함을 최소화해야한다. 적은 결함은 실행 중에 장애가 적다는 것을 의미한다. 결함 회피 기법은 **광범위한 컴파일러 검사를 허용하는 강한 자료형의 언어 사용과 포인터와 같이 오류가 발생하 기 쉬운 언어 구성 요소의 사용을 최소화**하는 것을 포함한다
결함 감지 및 정정	**검증 및 확인 프로세스는 프로그램이 운영을 위해 배치되기 전에 결함을 발견하고 제거하도록 설계**된다. 중대한 시스템은 배치 전에 가능한 한 많은 결함을 발견해 내고 시스템 이해 당사자들과 규제 기관에게이 시스템이 신뢰할 만하다는 것을 확신시키기 위해 광범위한 검증 및 확인을 필요로한다. **체계적인 테스팅. 디버깅, 정적 분석은 결함 감지 기법의 예**이다
결함내성	**실행 중에 결함이나 예상치 못한 시스템의 행동을 알아 내고 시스템 고장이 발생하지 않는 방식으로 관리되도록 시스템을 설계**해야한다. 내장 런타임 (runtime) 검사에 기반하는 간단한 방식의 결함 내성이 모든 시스템에 포함될 수 있 다. 높은 수준의 시스템 가용성과 신뢰성이 요구 될 때는 11.3 절에서 논의하는 결함 내성 시스템 아키텍처와 같은 특수한 결함 내성 기술이 사용될 수있다.
불행히도 3개의 기술들을 적용하는것이 항상 효과적인 것은 아니다.	

📜 가용성과 신뢰성

✔️가용성

- 주어진 시점에서 시스템이 운영 중이고 요청된 서비스를 제공할 확률

✔️신뢰성

- 주어진 환경에서 특정 목적을 위해 지정한 시간동안에 고장없이 운영될 확률
- 시스템 신뢰성은 절대적인 값은 아니다.
- 신뢰할 수 없는 사람이란? = 약속을 지키지 않는 사람
- **고장 이란?**
  - 명세서대로 준수해서 돌아가지 않는것
- **고장의 정의(명세서대로 준수해서 돌아가지 않는것)의 문제점**
  - 소프트웨어 명세서는 종종 불완전하거나 부정확하다
    EX) 바르샤바 공항사고
  - 시스템 개발자를 제외하고는 아무도 명세서를 읽지 않음

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️신뢰성 요구사항

- **바르샤바 공항에서의 사고**
  - 1993년 9월
  - 시스템은 요구사항 명세서대로 동작했으나 사고가 발생되었음
    - 활주로 면이 미끄러워 랜딩 기어가 지면에 닿지 않아 시스템은 비행기가 아직 날고있다고 판단하고 역추진 장치가 작동 하지 않음
    - 기능적 요구사항의 결함
  - 요구사항 명세서의 중요성을 알게 해준 사례임

- **요구사항 명세서의 두 유형**
  - **기능적 요구사항(기 복보)**
    - 시스템에 포함되어야 하는 **점검 및 복구 기능**과 **시스템 장애 및 외부 공격에 대한 보호 기능**을 정의한다.
  - **비기능적 요구사항(비 가신)**
    - 요구되는 시스템 전체 **신뢰성 및 가용성**을 정의한다.

✔️📌 비기능적 → **신뢰성 척도** ⭐322.p

신뢰성

- 온 디맨드 고장 확률 **(POFOD) → SIL**
  - **Probability Of Failure On Demand (POFOD)**
  - 시스템 서비스에 대한 요구가 시스템 장애를 일으킬 확률이다.
    EX) 어떤 요구가 있을 때 POFOD = 0.001 → 1/1000 확률로 고장발생한다는 의미

- 고장 발생 비율 **(ROCOF)**
  - 특정시간 간격 또는 시스템 실행 횟수에서 발생할 수 있는 시스템 장애의 확률

가용성	설명
0.9	시스템을 90%의 시간 동안 사용할 수 있다. 이는 24시간(1,440 분) 중에 144분 동안 시스템을 사용할 수 없다는 것을 의미한다.
0.99	24시간 중에 14.4분 동안 시스템을 사용할 수 없다.
0.999	24시간 중에 84초 동안 시스템을 사용할 수 없다.
0.9999	24시간 중에 8.4초 동안, 대략 1주일 중에 1분 동안 시스템을 사용할 수 없다.

323.p | 그림 11.4 가용성 명세서

가용성

- **가용성 척도 (AVAIL)**
  - AVAIL은 서비스에 대한 요구가 있을 때 시스템이 가동중일 확률이다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

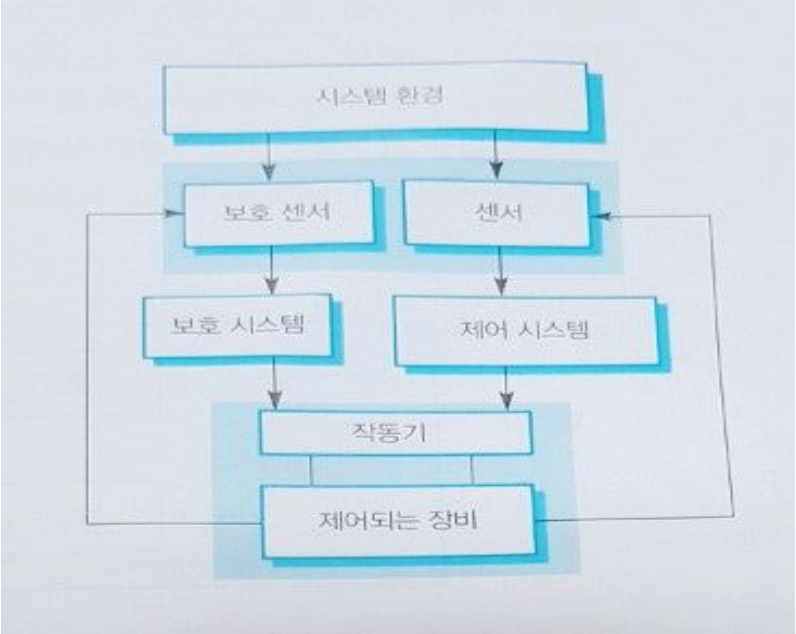
✔️기능적 신뢰성 요구사항

기능적 신뢰성 명세 ⭐326.p 중복 프로세스 검사 복구	
검사 요구사항	**부정확하거나 범위를 벗어난 입력이 시스템에 의해 처리되기 전에 감지되는 것**을 보장하기 위한 시스템 입력 검사들을 식별한다.
복구 요구사항	**장애가 생긴 후 시스템 복구를 도와주는 데 맞춰져 있다.** 이러한 요구사항은 보통 시스템 및 그 데이터의 사본을 관리하고 장애 발생 후 시스템 서비스를 복원하는 방법을 명시하는 것에 관련되어있다.
중복성 요구사항	**단일 컴포넌트 고장이 서비스의 완전한 손실로 이어지지 않는다는 것을 보장하는 시스템의 중복 기능들을 명시**한다
프로세스 요구사항	**개발 프로세스에서 좋은 실무 관행이 사용되는 것을 보장하는 결함 회피 요구 사항**이다. 명시된 실무 관행은 시스템 장애 횟수를 감소시켜야한다

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜결함내성 아키텍처

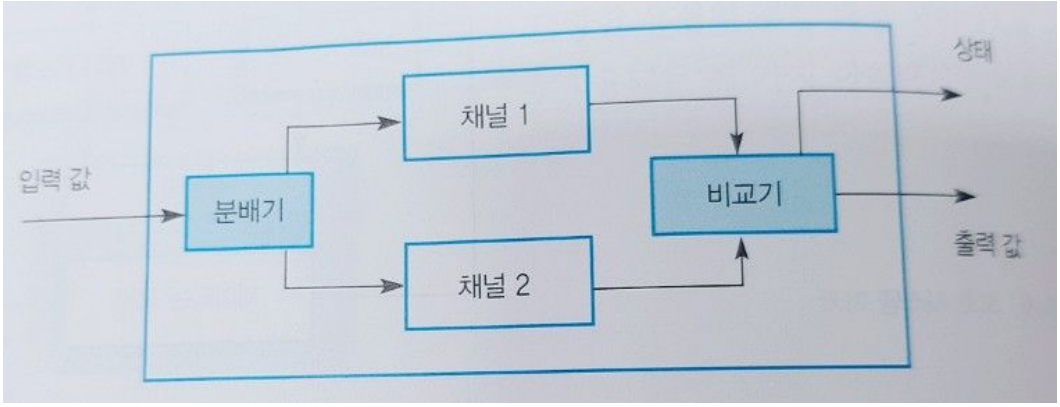
✔️보호 시스템 아키텍쳐 (protection system architecture) ⭐329.p



- 보호시스템은 다른 시스템과 관련되는 특별한 시스템이다.
- **보호 시스템은 독립적으로 자신의 환경을 감시한다.**
  - 보호 시스템은 제어되는 장치와 환경 모두를 감시한다.
- **보호 시스템은 잠재적으로 안전하지 않은 상태에서 안전한 상태로 시스템은 전환하는 데 요구되는 중요한 기능을 포함한다.**
- 운영을 감시하고 위급한 상황이 발생했을 때 시스템이 안전한 상태로 전환되는 것을 보장하는 것이다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️자기점검 아키텍처



- 자신의 동작을 점검하고 만약 문제가 발견되면 조치를 취하도록 설계된 시스템 아키텍처다.

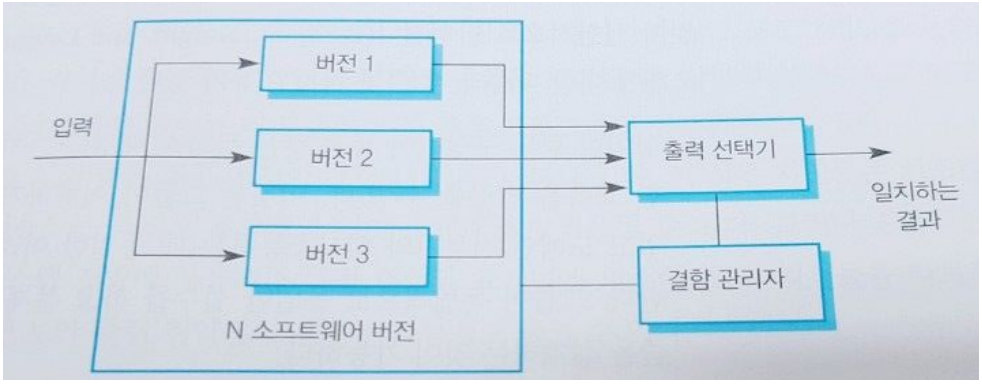
- 멀티 버전 프로그래밍이 소프트웨어 중복성과 다양성을 제공하기 위해 사용된 시스템의 예이다.

- 각각의 채널에 사용되는 하드웨어가 다양하다.
    - 다양한 제조사로부터 칩셋이 공급이 가능.

- 각각의 채널에 사용되는 소프트웨어가 다양하다.
    - 동일한 소프트웨어 오류에대해 대처가능.

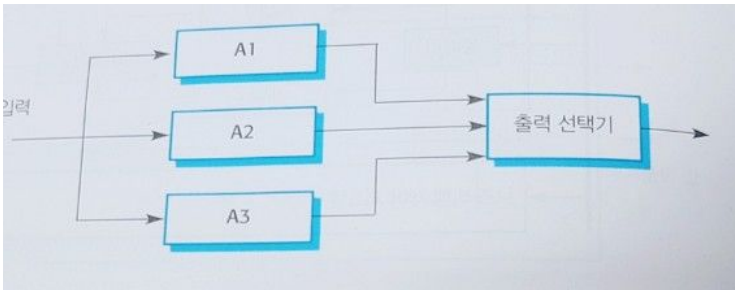
제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️ N버전 프로그래밍



- 동일한 소프트웨어 시스템이 공통의 명세서를 사용하여 다수의 팀에 의해 구현된다.
- 매우 높은 소프트웨어 개발 비용
    - → 다양한 버전의 소프트웨어를 개발하기 위해 여러 개의 팀이 필요

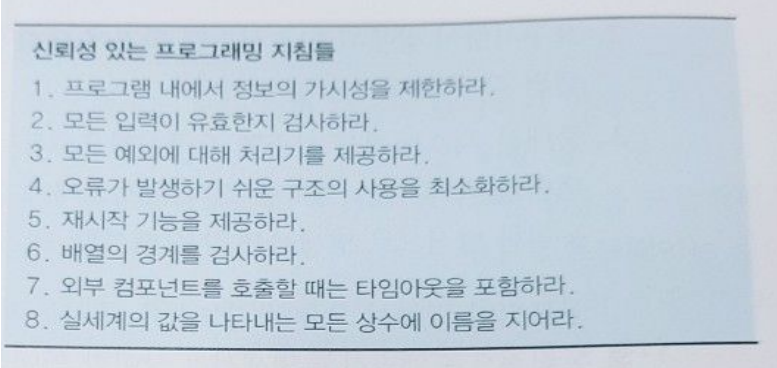
- 삼중 모듈 중복



제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜📌결함신뢰성을 위한 프로그래밍 (⭐335.p)

- 신뢰성 있는 프로그래밍 지침들



✔️ 지침1 : 프로그램 내에서 정보의 가시성을 제한하라.

✔️ 지침2 : 모든 입력이 유효한지 검사하라.

✔️ 지침3 : 모든 예외에 대해 처리기를 제공하라.

✔️ 지침4 : 오류가 발생하기 쉬운 구조의 사용을 최소화하라.

✔️ 지침5 : 재시작 기능을 제공하라.

✔️ 지침6 : 배열의 경계를 검사하라.

✔️ 지침7 : 외부 컴포넌트를 호출할때는 타임아웃을 포함하라.

✔️ 지침8 : 실세계의 값을 나타내는 모든 상수에 이름을 지어라.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

😎안전성 공학

📜안정성 달성

✔️ Safety Achievement (⭐ 354.p)

- Hazard avoidance **(위험 회피)**
  - Hazard = 위험요인
  - 위험을 피할 수 있도록 시스템을 설계한다.

- Hazard detection and removal **(위험 감지 및 제거)**
  - 사고가 발생하기 전에 위험을 감지하고 제거할 수 있도록 시스템을 설계한다.

- Damage limitation **(손실제한)**
  - 시스템은 사고피해를 최소화하는 보호 기능을 포함할 수 있다.

위험 (risk) = 발생빈도 * 심각도

✔️안전성 달성을 하는데 있어서 2가지 역할

- 시스템이 소프트웨어로 제어되므로 소프트웨어가 내린 결정과 이후의 행동이 안전 성 중심이다. 따라서 소프트웨어 동작이 시스템의 전체 안전성에 직접적으로 관련된다

- 시스템의 다른 안전성 중심 컴포넌트들을 검사하고 감시하기 위해 소프트웨어가 광범위하게 사용된다 예를 들어 모든 항공기 엔진 컴포넌트들은 컴포넌트 장애의 초 기 징후를 찾는 소프트웨어에 의해 감시된다.이 소프트웨어에 장애가 발생하면 다 른 컴포넌트들이 장애로 인해 사고를 일으킬 수 있기 때문에이 소프트웨어는 안전 성 중심이다.

✔️소프트웨어가 안전하지 않을 수 있는 4가지 이유

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

📜안전성 요구사항

✔️위험 식별 ⭐328.p

- 위험 식별 프로세스는 시스템을 위협할 수 있는 위험들을 식별한다. 이러한 위험은 위험 기록부에 기록될 수 있다.
    - 위험 기록부 : 안전성 분석 및 평가를 기록한 공식 문서

📘📌 Hazard-driven analysis

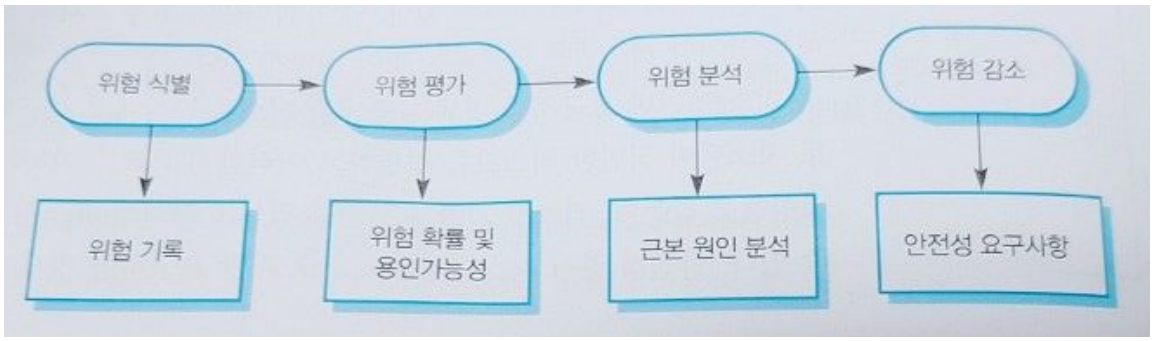


그림 12.2 | 위험주도 요구사항 명세화

- 위험원 식별
    - hazard log

- 위험원 평가
    - 발생빈도

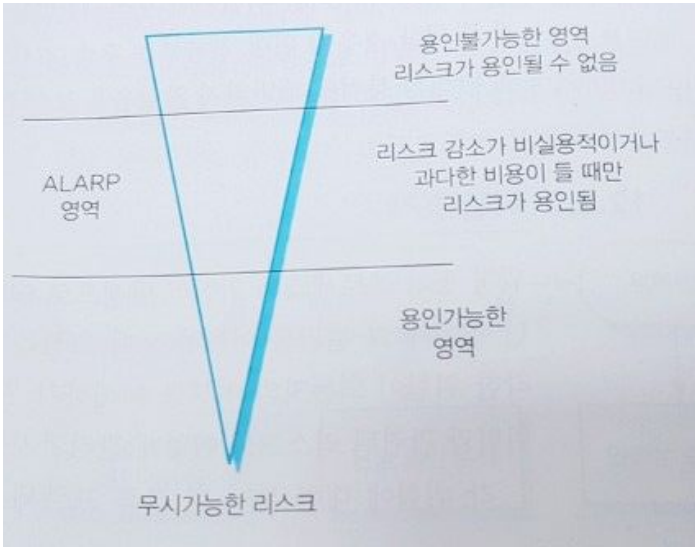
제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

- 📌 위험원 분석
  📣 **| 시험에 나옵니다.**
  - root case
    - 📌 **FTA**
      - **결함 트리 분석(Fault Tree Analysis)**
      - top down
      - **tree 형태**
      - https://grapevine9700.tistory.com/222
      - **안전필수 시스템(safety-critical systems)의 신뢰성(reliability) 및 안전성(safety)을 분석하는 기법 중 하나이다.**
    - 📌 **FMEA**
      - **고장 형태 영향 분석**
        (Failure Mode & Effect Analysis, FMEA)
      - bottom up
      - **표형태**
      - https://ergonomics.tistory.com/18
- 안전 대책 (위험 감소)
  - Sw 안전 요구사항
    - SSRS 4단계

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️위험 평가

- 위험 평가 프로세스는 어떤 위험들이 가장 위험한지와 가장 발생 가능성이 높은지를 결정한다.



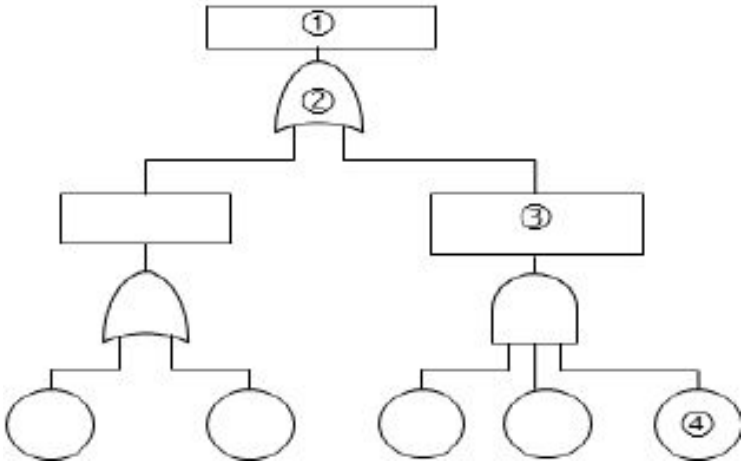
- 위험 평가에 사용되는 3가지 리스크 유형
  - 용인불가능 리스크
  - 일어날 확률이 적은(ALARP) 리스크
  - 용인가능한 리스크

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

✔️위험 분석

- 위험의 발생으로 이어질 수 있는 사건들을 식별하는 근본 원인 분석 프로세스이다.
    - **안전성 중심 시스템에서 위험의 근본 원인을 찾는 프로세스이다.**
    - **목표 :** 위험을 초래할 시스템 고장을 야기하는 사건이나 사건의 조합을 찾는것

- **위험 분해나 분석방법**
    - 결함트리분석 기술을 사용
        - **특징**
            - 전문적인 특정 분야의 지식 없이도 이해하기 상당히 쉽다.
            - 트리의 루트에 위험을 놓고 그 위험에 이를 수 있는 시스템 상태를 찾아낸다.
            - 각 위험에 대해 그 위험의 가능한 원인을 찾기 위해 역방향으로 작업을 수행한다.



✔️위험 감소(=리스크 감소)

- 이 프로세스는 위험 분석의 결과에 기초하며 안전성 요구사항의 식별로 이어진다.

제목: 😀 부제목:😎 인덱스:📜 교수님:📣 쪽집게:📌 중요: 책참고: ⭐ 📖 솔루션:💡 📘📒✔️

😎보안성 공학

😎복원성 공학