

1. Compare sociotechnical systems to conventional technical computer-based systems.

: 전형적인 기능 중심의 시스템에 비하면 고려 사항을 추가해야 하는, 그러나 시스템의 가치를 높이기 위해서 중요한 것.

- system

- 공통의 목적을 달성하기 위해 같이 동작하고 서로 관련이 있는 컴포넌트들의 목적을 갖는 컬렉션
- 시스템은 소프트웨어, 기계 전기 전자적인 하드웨어를 포함하며, 사람에 의해 작동함
- technical system들은 시스템의 다른 컴포넌트들에 의존적이다
- system component들의 속성과 행동은 서로 뒤섞인 불가분의 것이다.

① Technical computer-based systems

: 시스템은 하드웨어와 소프트웨어를 포함하지만, operator나 operational processes는 시스템에 포함시키지 않는다. 즉 시스템은 self-aware하지 않음

ex) word processor used to write a book.

② Socio-technical systems

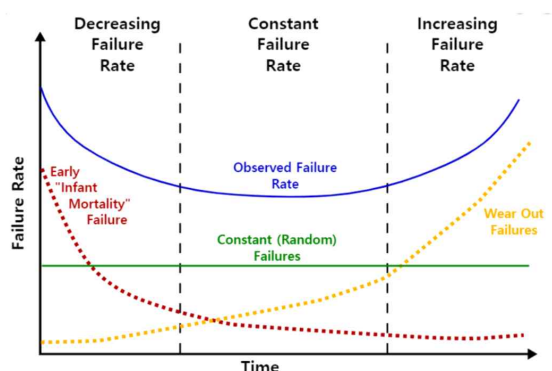
: technical system을 포함하는 시스템으로 technical system을 이용하는 technical system과 상호작용하는 프로세스와 사람을 포함함. Socio-technical systems은 조직의 정책과 규칙(organisational policies and rules)에 의해 통제된다.

ex) publishing system to produce a book

2. Assume the difference between a hardware failure and a software failure. Note that a hardware failure is represented by a bathtub curve.

: software failure는 시간이 지날수록 시스템이 진정화되어 failure가 떨어지지만, hardware failure는 초반에는 failure가 줄어들지만 안정적으로 가다가 어느 시간을 기점으로 failure가 급격히 증가함. 왜냐하면 hardware component의 수명이 다 되기 때문이다.

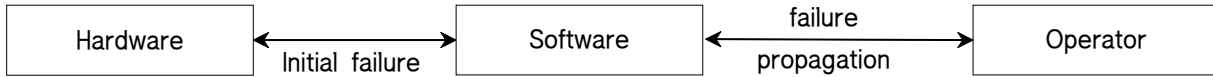
software의 lifetime은 없다고 계속 0에 가깝게 수렴하진 않는다. software reliability는 신뢰성을 설명하는 중요한 특성으로 시스템을 구성하는 컴포넌트가 각각 reliable하다고 해도, 시스템이 reliable하다고 이야기 할 수 없다. 따라서 어떤 컴포넌트끼리 영향을 미치는 relationship을 예측하도록 노력해야 한다. 하지만, software/hardware/operator reliability가 서로 영향을 미치므로 다른 쪽에 문제가 생겼을 때 영향을 받을 수 있다.



예측하도록 노력해야 한다. 하지만, software/hardware/operator reliability가 서로 영향을 미치므로 다른 쪽에 문제가 생겼을 때 영향을 받을 수 있다.

+ evolution 하면서 다시 다른 sub-software에 영향이 어디까지 되는지 모름

3. Discuss about failure propagation between hardware, software and operator.



: hardware에서 시작한 failure는 software에서 operator까지 propagation되었다가 다시 돌아오게 된다

센서의 고장으로 software 개발 필요 ex) 20~30도에서 측정하는 센서 → 200도 측정 시 이에 상응하는 software 필요

4. Discuss and remember about the system engineering process and the relationships between conceptual design, procurement, development, and operation.

① conceptual design: functional requirement는 추상적으로 구상하고, non-functional requirement는 high-level로 구상함

팀의 능력 / 환경에 따라 구매 or 자체 제작 결정 (살수 있음 사라)

② procurement(acquisition) : 어떤 시스템 component를 구매 할 수 있을지 결정함 구매 (COTS). 외주, 자체제작

③ development : 시스템 개발해야할 곳을 개발하고, 통합하여 user의 니즈에 맞게 training됨

④ operation : 실제 operational 환경에 배포됨

↑  
opensource 이용  
(BUT. 어려움)

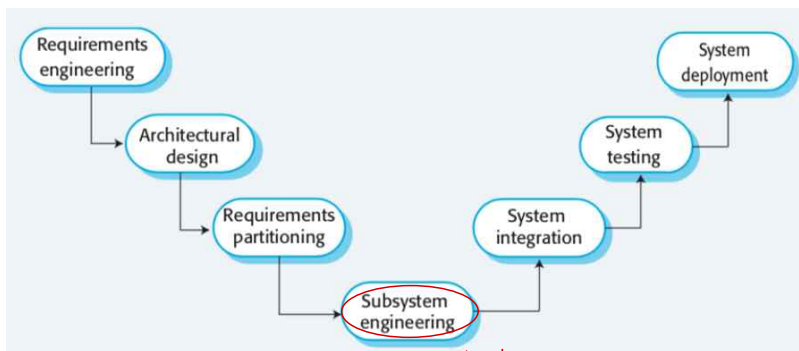
사용법 / 기능 → 문서화되어 있을 때 가져와서 쓸 수 있음 (재사용)

5. What do you think is necessary for parallel development of large systems?

팀간의 공통적 단어를 쓸 수 있도록 충분한 토의 (시설 상 토론 어려움)

architecture design 이 있어야 parallel design, integration 할 때 중요

6. Discuss and remember the system development process from requirement development to system deployment.



software 관점에서

requirement → design → development → integration

7. Discuss why procurement decisions affect system functionality, such as reliability.

- off-the-shelf system에서는 procurer는 security나 dependability에 영향을 적게 받음
- custom system에서는 큰 영향을 받음
- 이는 이미 기존에 만들어진 system은 이런 단계를 미리 고려하였지만, 우리가 개발해야할 시스템은 처음부터 우리가 생각해야하기 때문임

구매한 component의 reliability가 전체 reliability에 영향을 받음

비용 부족으로 testing 부족으로 dependability ↓

8. Discuss how the various activities of the system development process affect system dependability. 과 testing의 관계

- dependability : 시스템에 대한 사용자의 신뢰도로 이게 달성되기 위해서는 시스템이 available/reliable/securable/safe 해야 함
- development와 dependability는 trade-off 관계임
- human error로 인해 system에 결함이 발생할 수 있음
- 제한된 예산으로 test나 validation을 제한하거나, 비싼 개발 기술과 hardware를 사용하지 못할 수 있음
- 배포할 시에 문제가 있는 경우는 system과 operational environment가 맞지 않을 수 있음

과 requirement의 관계

문서화가 잘 되어 있지 않다면 목표가 명확 x  
요구사항 못 맞춰서 dependability ↓

과 evolution의 관계

(requirement) 목표가 명확 x → 유지보수 어려움

why? candidate의 선택의 이유가 중요

→ 의사결정 요소를 대체할 수 있는지 판단 가능

9. Discuss one or more examples of the various barriers to protect human error.

- 삭제하기와 같은 명령을 주었을 때, 정말 삭제할 것이냐고 물어보는 과정

10. Discuss and summary about why the evolution is costly.

새로운 표준, 고객의 요구변경, 기술환경 변화 등 evolution 되어야 하는 이유는 많음

- sub-system의 interface가 다르므로 예상치 못한 문제가 일어날 수 있음
- original design decision을 한 이유/근거가 기술이 잘 되어 있지 않아서 이를 조사하기 위해서 비용 소모
- 많은 변화 때문에 system structure은 이미 많이 변화되어 있어 다른 sub-system에 어떻게 영향을 끼칠지 모름

#### MY QUESTION

1. Technical computer-based systems에서 소프트웨어는 포함하지만, operational processes는 시스템에 포함시키지 않는다. 라고 적혀있는데 operational processes도 software의 한 종류가 아닌지?

: 여기서 process는 software가 아니라 특정 고객을 대상으로 특정 서비스의 제품을 생산하는 활동이나 태스크의 구조, 관계에 대한 집합임

2. dependability안에 reliability가 포함되어 있는 것 같은데 정확히 두 개가 어떻게 다른지?

dependability : The ability to deliver service that can justifiably be trusted.

reliability : The ability of the system to deliver services as specified.