

시민 안전은 누가 책임지나? Software-Defined Safety

배 두환
전산학부
한국과학기술원

우리 일상속의 소프트웨어:

Ubiquitous Software

- "Software is everywhere"
 - 보이지는 않지만 많은 SW와 함께 살고 있다.
- 자동차
 - 고급 승용차: 1억 lines of code
 - 자율 주행: SW를 통한 시스템 제어
 - 전기/수소차: 안전성 확보를 위한 SW 시스템
- 스마트폰, 정보가전,..
- 고속철, 지하철, 항공기, 선박, 무기 체계,...
- 병원, 경찰서, 행정 기관, 교육 기관,...

우리 사회는 얼마나 안전한가?(1/4)

- 우리 나라의 안전 관련 통계 (국토매일, 2019년 3월 15일)
 - 대한민국 국민 1인당 실질 국내총생산 기준 세계 11위(2016)
 - 10년간 GDP는 약 28.6% 증가, 반면에, '안전' 지수는 22.6% 증가
 - 다중밀집시설 대형 화재, 붕괴 대형 사고, 지하철 사고, 산불 등 총 66건, (2007-2016)
 - 6.6회/년, 사망자 835명

 대부분의 사고는 인재(人災)

우리 사회는 얼마나 안전한가?(2/4)

- 2017년, [충북 제천 대형 화재](#)
 - 29명 사망,
 - 13명의 소방 인력 출동, 실제 불을 끌 수 있는 인원 4명
 - 18분간 무전 교란 (2층 자동 출입문 앞에서 20명 사망)
- 2018년, Grenfell Tower Fire ([Grenfell Tower report](#))
 - 72 명 사망
 - 인슐레이션 소재 문제
 - 재난 대응의 문제는?
 - 화재 대피 요령
 - 'Stay put'
 - 비상등 표시 부적절
 - 통신 불량



우리 사회는 얼마나 안전한가?(3/4)

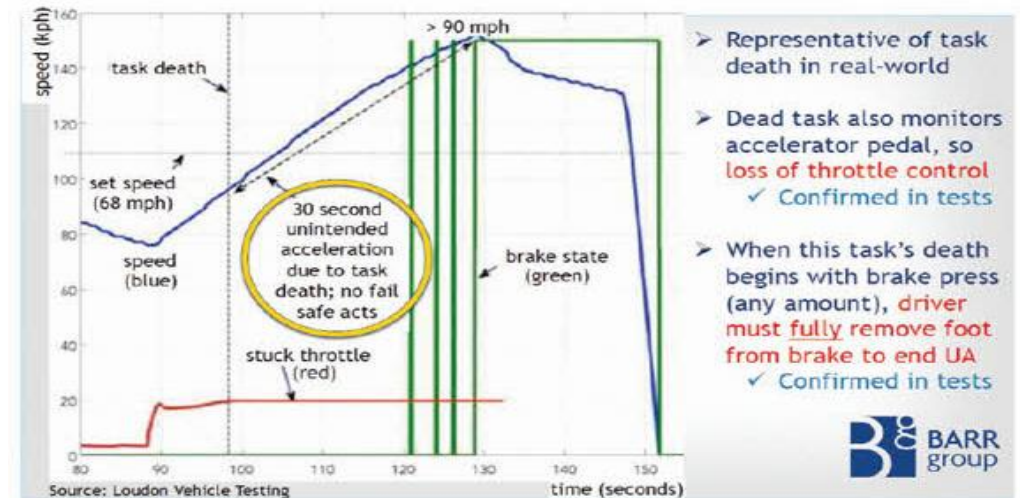
• 자동차

- 토요타 Camry 자동차의 급발진 사례: ([automotive 2014년 9월호](#))

- 미국의 Barr 그룹에서 SW 시스템 오류로 인해서 급발진 가능성 재현, 보고서 제출
- FMEA 불안전, 운영체제 코드에 대한 검증 부족, 코드 표준 미사용 등

- 자율 주행 자동차 사고, ...

- Google, Tesla, ...



우리 사회는 얼마나 안전한가?(4/4)

- 보잉 737 맥스 항공 사고
 - 346명 사망
 - 개발자/관리자의 실수
 - SW is used as 'band aid'.
 - 인증 실패 등



어떻게 우리 사회를 더욱 안전하게?

- Grenfell , 충북 제천 화재 사례:
 - 기능 안전: 안전 사고 예방/대응을 위한 SW
- 토요타 캠리 사례
 - SW 결함 최소화 노력: 안전한 SW
- 보잉 737 맥스
 - SW 시스템 설계, 프로젝트 관리, 안전 인증: SW Engineering



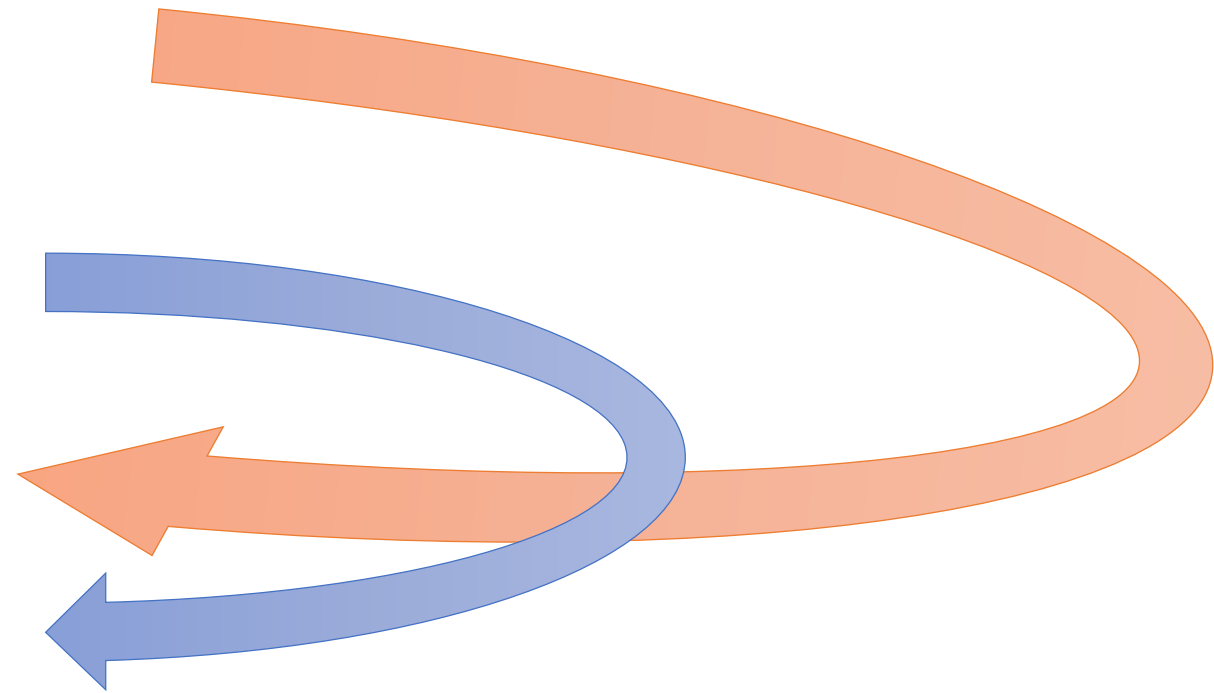
Software-Defined Safety

Software-Defined Safety

- 안전 중심의 소프트웨어 공학 실무
 - 현황: 더 많은 기능을 사람이 아닌 SW 시스템이 제어
 - SW 개발에서의 인재로 인한 재난 사고 가능성 방지 노력.
- 제조 산업의 새로운 부가 가치 창출
 - 현황: 기본적인 SW 역량만으로는 경쟁력 부족
 - Safety를 통한 차별화 전략

➡ Safety for Software

➡ Software for Safety



SW Defined Safety 역량 확보를 통한 경쟁력 제고

- 기본적인 SW 역량
 - 더 좋게(Better), 더 빠르게(Faster), 더 싸게(Cheaper)



- 4차 산업 혁명에 요구되는 SW 역량
 - 기본적인 SW 역량 + 'X' ← Safety
- 기존 산업의 차별화/가치화 측면에서의 Safety 의 역할 중요