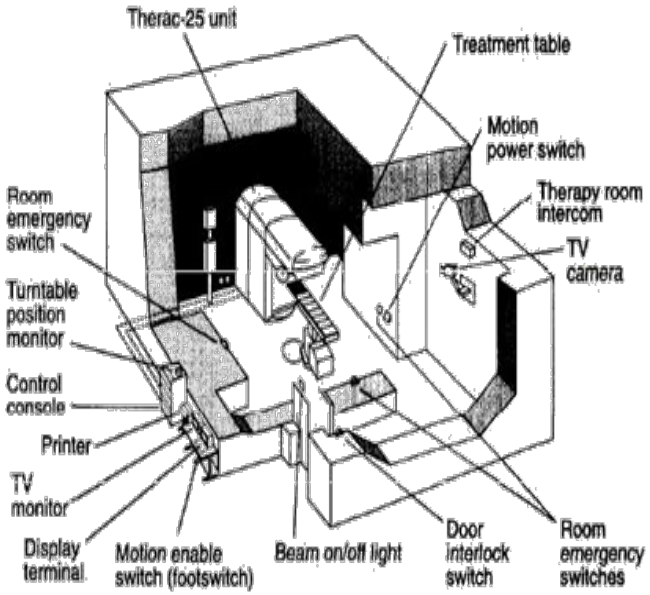
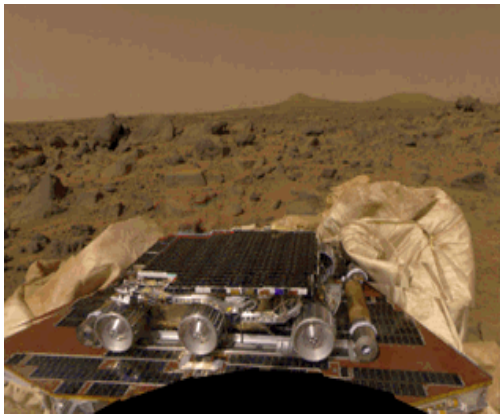
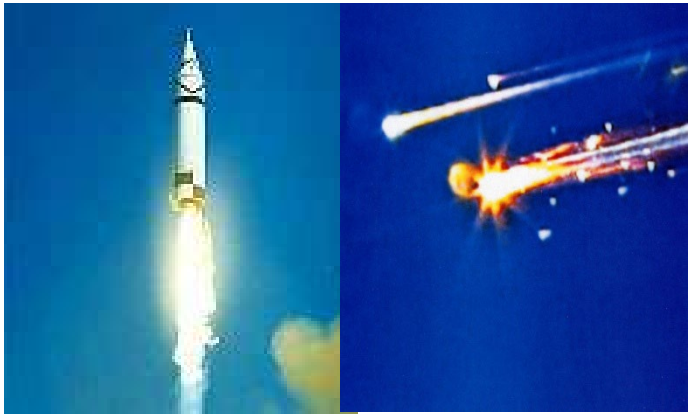
Current Practice for SW Development

- **SW developers have to follow systematic disciplines for building and analyzing software with high quality**
  - This class focuses on the analysis activities

Safety Problems due to Poor Quality of SW

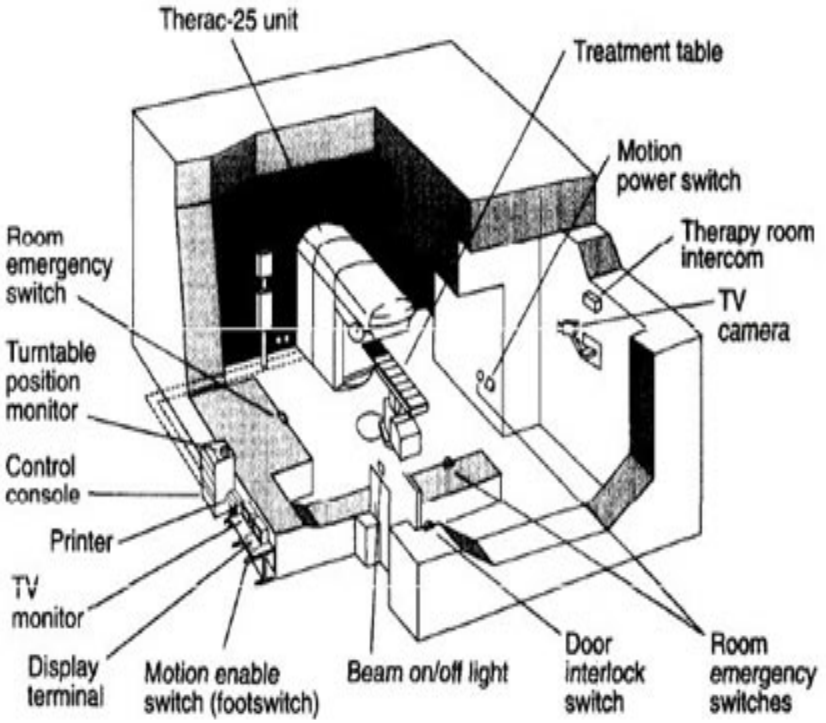






Tragic Accidents I

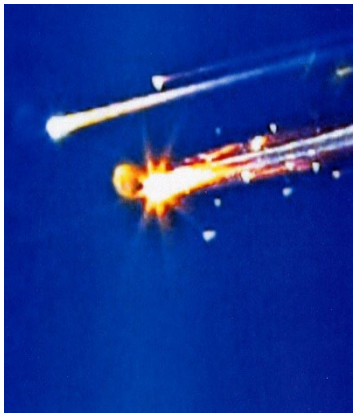
- The Therac-25 Story
  - Between June 1985 and Jan 1987, a computer-controlled radiation therapy machine, called the Therac-25, massively overdosed six people
    - software coding error



- **http://sunnyday.mit.edu/papers/therac.pdf**

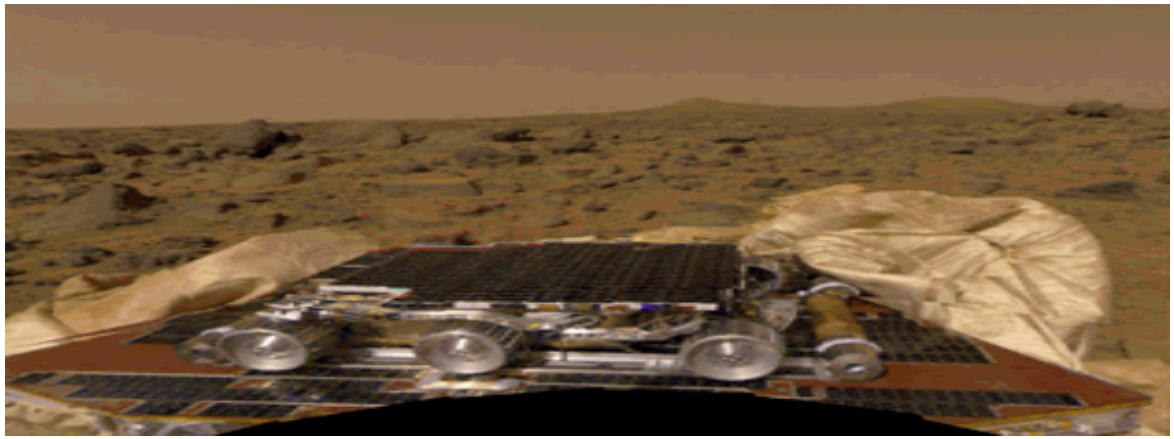
Tragic Accidents II

- Ariane 5
  - "On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure…The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information …This loss of information was due to specification and design errors in the software of the inertial reference system."
    - Floating number conversion problem
    - http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html



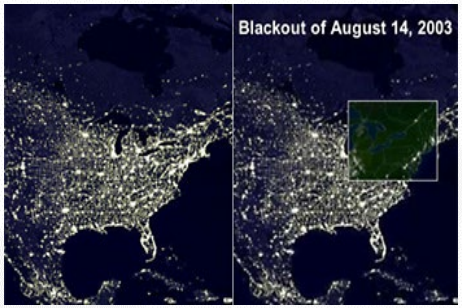
Tragic Accidents III

- NASA Mars Pathfinder (1997)
  - Priority inversion problem led to a system reset and a one-day delay in retransmission of data which wasted valuable mission time.
  - http://www.cis.ksu.edu/~hatcliff/842/Docs/Course-Overview/pathfinder-robotmag.pdf



Social and Economic Loss due to High Complexity of SW

Although most areas of modern society depend on SW,
reliability of SW is not improved much due to its high complexity



(2003) US & Canada blackout
- 7 states in US and 1 state in Canada suffered 3 days electricity blackout
- Caused by the failures of MISO monitoring SW
- 50 million people suffered and economic loss of 6 billion USD

(2010s) Toyota sudden unintended acceleration
- 89 people died since 2002
- SW bugs detected in 2012
- Fined 1.2 billion USD in 2014

TOYOTA

(2018-2019) Boeing 737 MAX accidents
- 346 people died in 2 accidents
- SW bugs detected in 2019
- Boeing 737 MAX is banned all over the world

BOEING

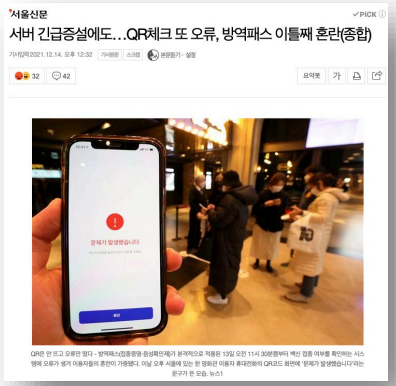
Software Failures

July 19, 2024

MS Cloud failure due to wrong update to CrowStrike security software

December, 2021

Failure of the QR check-in system



March 10, 2017

Unable to unlock Tesla cars due to the problem of service APIs

July 2, 2015

Casualty at the Volkswagen factory due to faulty movements of robot arms

Accidents of Autonomous Vehicle

주요 자율주행차 사고

일시	관련 기업	사고 내용
2016년 2월 14일	구글	자율주행차의 과실로 시험 주행 중 버스와 접촉사고
2016년 5월 7일	테슬라	대형 트레일러와 충돌, 첫 운전자 사망사고
2017년 11월 8일	나브야	자율주행셔틀, 실주행에서 트럭과 접촉사고
2018년 3월 18일	우버	교차로 건너던 행인 치어 첫 보행자 사망사고
2018년 3월 23일	테슬라	고속도로에서 중앙분리대를 들이받아 운전자 사망

지난달 23일 발생한 테슬라 자율주행 차량의 사망사고 현장. 지난해 9월 일어난 중앙분리대 충돌사고처럼 오전 역광(아래 사진)이 사고 원인이라는 주장이 제기됐다. [미 폭스TV·ABC 방송 캡처]

[출처: 중앙일보] "테슬라 사고, 태양 역광 탓" … 자율주행차 또 날씨 오작동

Bugs of Autopilot SW

연합뉴스

도로에 점 3개 찍었더니..테슬라 자율주행차, 中실험서 역주행

입력 2019.04.02. 10:31 댓글 338개

도로 위 작은 점 칠해 자율주행시스템 속여..테슬라 "실제 일어날 문제아냐"

(서울=연합뉴스) 정성호 기자 = 중국의 사이버보안 연구소가 전기차 테슬라의 자율주행 시스템을 속여 이 차가 반대편 차선을 역주행하도록 한 것으로 나타났다.

중국의 게임·인터넷 기업 텐센트 산하 '킨 시큐리티 랩'은 테슬라 전기차의 오토파일럿 시스템에 대한 실험에서 이처럼 차가 역주행하도록 했다고 밝혔다고 블룸버그 통신과 경제 매체 비즈니스인사이더가 1일(현지시간) 보도했다.

이 연구소는 실제 도로 위에 세 개의 작은 점을 칠했고, 그 결과 테슬라의 전기차가 왼편에 있던 반대편 도로로 옮겨 역주행하도록 하는 데 성공했다고 밝혔다.

도로의 교차로 지점에 작은 점을 표시하자 테슬라의 차가 이를 오른쪽 차선으로 인식하고 왼쪽으로 방향을 틀어 주행했다는 것이다.

테슬라의 전기차. [AFP=연합뉴스 자료사진]

The weakness of Data Driven AI/Machine Learning SW (adversarial example)

input image
(street sign)

Anomaly
(bird nest)

Boeing 737 MAX Crashes due to SW Bugs

참화 부른 보잉의 '늑장대응'…이제 와서 "열흘내 업그레이드"(종합)

송고시간 | 2019-03-16 06:54

NYT "보잉, 작년말까지 업그레이드 약속"…셧다운發 업무지연 연관성도 주목

(뉴욕=연합뉴스) 이준서 특파원 = 미국 항공기 제작업체 보잉이 전 세계적으로 운항중단 조처가 내려진 '보잉 737맥스(Max)' 기종에 대해 10일 이내 '소프트웨어 업그레이드'에 들어갈 예정이라고 AFP통신이 15일(현지시간) 보도했다.

문제로 지목된 소프트웨어는 '조종특성 향상시스템'(MCAS·Maneuvering Characteristics Augmentation System)이다. 난기류 상황에서 항공기의 급하강을 막아주는 일종의 운항정지 방지 시스템이다.

구체적인 원인 분석은 이뤄지지 않았지만, 4개월여 사이에 재발한 '737맥스 8' 기종의 추락 참사는 MCAD와 무관치 않은 것으로 분석된다.

"보잉, 737 맥스 조종제어 소프트웨어 대폭 수정 중"

SBS이혜미 기자

입력 : 2019.03.13 12:54 | 수정 : 2019.03.13 12:54

미국 보잉사가 안전성 우려가 불거진 737 맥스 기종 전반에 대해 조종제어 소프트웨어를 대폭 수정하고 있다고 월스트리트저널이 보도했습니다.

소프트웨어 수정은 지난 주말 에티오피아 여객기 추락 사고가 발생하기 전부터 진행해 온 것으로 지난해 10월 같은 기종인 인도네시아 라이언에어 여객기가 추락한 데 따른 것입니다.

미 항공당국은 다음 달 말까지 수정작업이 마무리될 것으로 예상하고 있다고 신문은 전했습니다.

보잉의 최신기종을 둘러싼 안전성 논란이 커지면서 보잉의 주가도 이틀째 추락했습니다.

SOFTWARE CAUSES OF MEMORY CORRUPTION

Type of Software Defect	Causes Memory Corruption?	Defect in 2005 Camry L4?
Buffer Overflow	Yes	Yes
Invalid Pointer Dereference/Arithmetic	Yes	Yes
Race Condition (a.k.a., "Task Interference")	Yes	Yes
Nested Scheduler Unlock	Yes	Yes
Unsafe Casting	Yes	Yes
Stack Overflow	Yes	Yes

Static analysis falls short of detecting such complex bugs accurately
- High false negatives
- High false positives

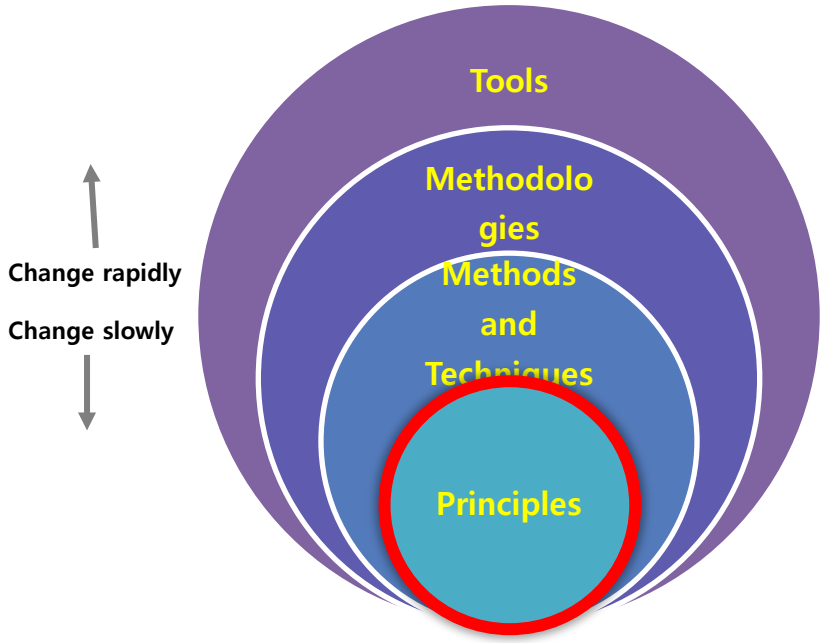
⇒ Systematic and dynamic analysis (i.e. automated sw testing) is MUST for high quality SW

BARR group

What is Software Engineering?

- "Software engineering is the form of engineering that applies the principles of computer science and mathematics to achieving *cost-effective solutions* to software problems" [CMU-SEI]
- "Software Engineering : (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1)." [IEEE]

Software Engineering Principles



Relationship between principles, techniques, methodologies and tool

[GJM02] Ghezzi et al., Fundamentals of Software Engineering

- **Principles**
  "General and abstract statements describing desirable properties of software process and product"
- **Methods**
  "General guidelines that govern the execution of some activity; they are rigorous, systematic, and disciplined approaches"
- **Techniques**
  More technical than methods and have more restricted applicability
- **Methodologies**
  Sets of methods and techniques selected for solving a problem
- **Tools**
  Tools to support the application of methodologies

Software Engineering Principles are for …

- Understanding what to develop
- Developing software the right way
- Developing the right software

by effectively collaborating with stakeholders, including customers, users, and developers.

https://www.linkedin.com/pulse/go-digital-why-diverse-inputs-multi-stakeholder-feedback-chris-leong/

Topics to be Covered

- **Understanding what to develop**
  - **Introduction** to Software Engineering
  - Software **Requirements** Engineering
- **Developing software the right way**
  - Process Models: Heavy-weight process, Prototyping, **Agile**, **DevOps**,..
  - Basics of **UML** (Unified Modeling Language)
  - SE Principles: **Separation of Concerns**, **Abstraction, Modularity**, etc.
  - Software Design: **Architectural** Patterns, **Resilient Design** Patterns
- **Developing the right software**
  - Software **Quality**: Correctness, Robustness, Safety, Security, etc.
  - Software Testing: **Testing overview**, **Black-box** and **White-box** testing

Readings

- [PrMa20] Roger S. Pressman, and Bruce R. Maxim, *Software Engineering – A Practitioner's Approach*, 9th Ed., Mc Graw Hill, 2020 (ISBN-13: 978-1259872976)
- [Som11] Ian Sommerville, *Software Engineering* (9th edition), Addison-Wesley, 2011
- [GJM02] Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, *Fundamentals of Software Engineering* (2nd edition), Pearson, 2002
- [LeWi99] Dean Leffingwell and Don Widrig, *Managing Software Requirements: A Unified Approach*, Addison-Wesley, 1999 (ISBN: 0-201-61593-2)
- [Coc01] Alistair Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001
- …