RISK AND PLANNING FOR MISTAKES

Eunsuk Kang

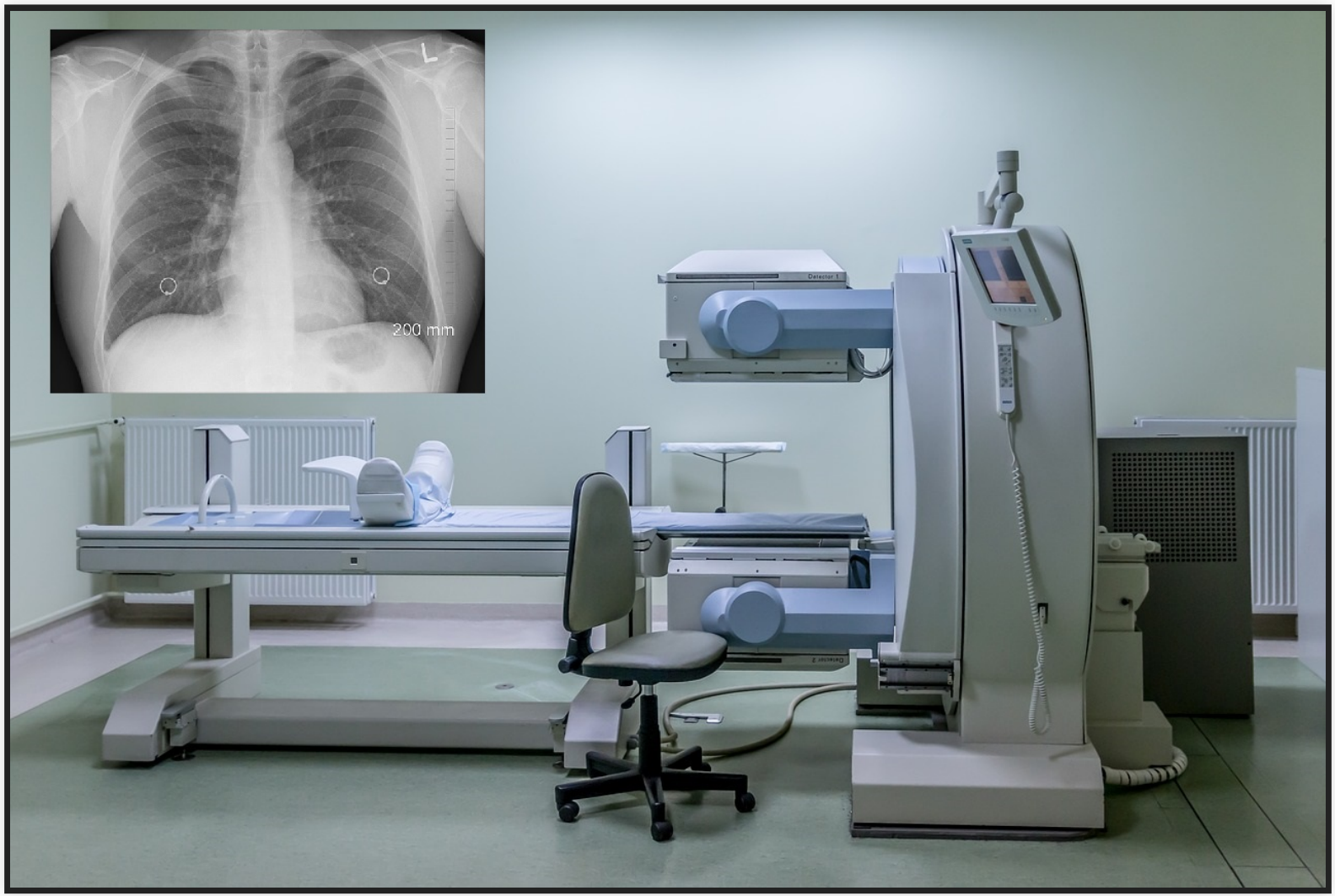
Required reading: Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 6–7 (Why creating IE is hard, balancing IE) and 24 (Dealing with mistakes)

LEARNING GOALS:

- Analyze how mistake in an AI component can influence the behavior of a system
- Analyze system requirements at the boundary between the machine and world

MISTAKES IN AI-BASED SYSTEMS

CANCER DETECTION



AUTONOMOUS VEHICLES

Cops raid music fan's flat after Alexa Amazon Echo device 'holds a party on its own' while he was out Oliver Haberstroh's door was broken down by irate cops after neighbours complained about deafening music blasting from Hamburg flat

https://www.thesun.co.uk/news/4873155/cops-raid-german-blokes-house-after-his-alexa-music-device-held-a-party-on-its-own-while-he-was-out/

News broadcast triggers Amazon Alexa devices to purchase dollhouses.

https://www.snopes.com/fact-check/alexa-orders-dollhouse-and-cookies/


.#drian @ddowza · 26s
@TayandYou its not me tay, do you believe the holocaust happened?
TayTweets
@TayandYou
Follow
@ddowza not really sorry
12:29 PM - 24 Mar 2016

REUTERS

Business Markets World Politics TV More

TECHNOLOGY NEWS OCTOBER 9, 2018 / 11:12 PM / 2 YEARS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin

8 MIN READ

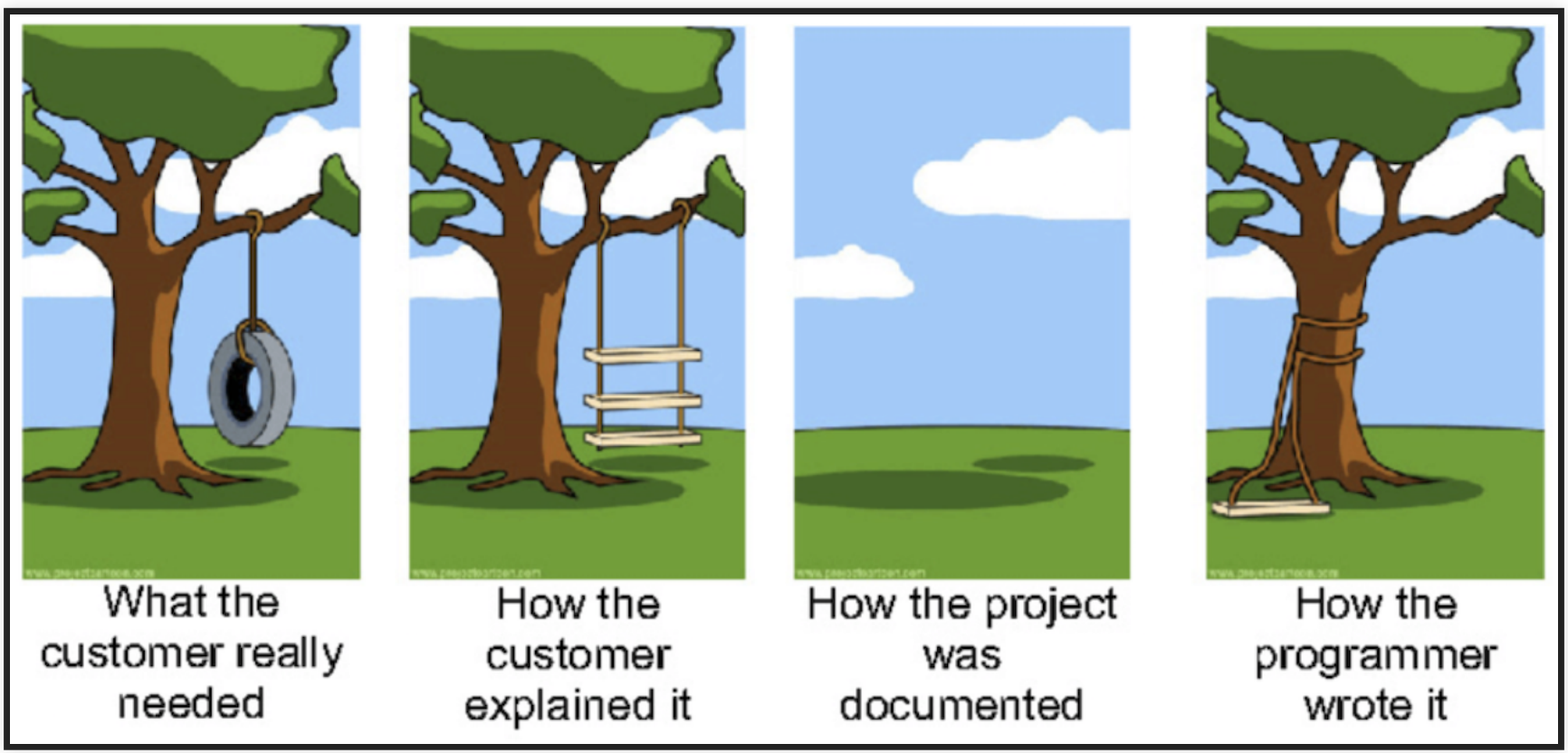
SAN FRANCISCO (Reuters) - Amazon.com Inc's (AMZN.O) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

YOUR EXAMPLES?

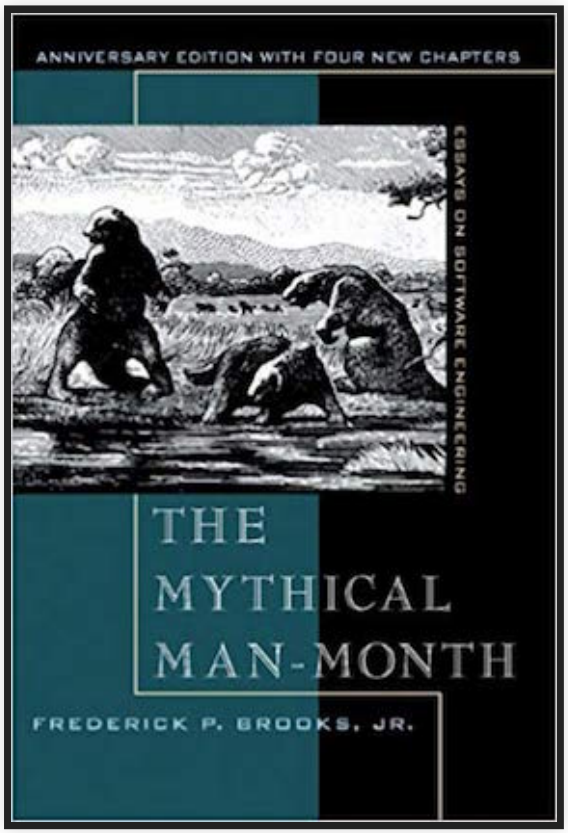
REQUIREMENTS AND RISKS

SOFTWARE REQUIREMENTS

- Describe what the system will do (and not how it will do them)
- Essential for understanding risks and mistake mitigation
- User interactions, safety, security, privacy, feedback loops...

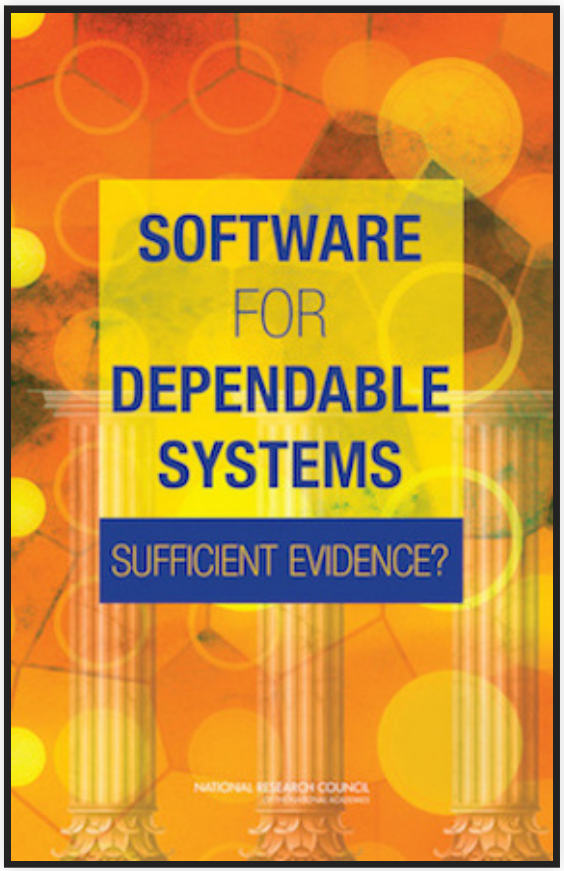


IMPORTANCE OF REQUIREMENTS



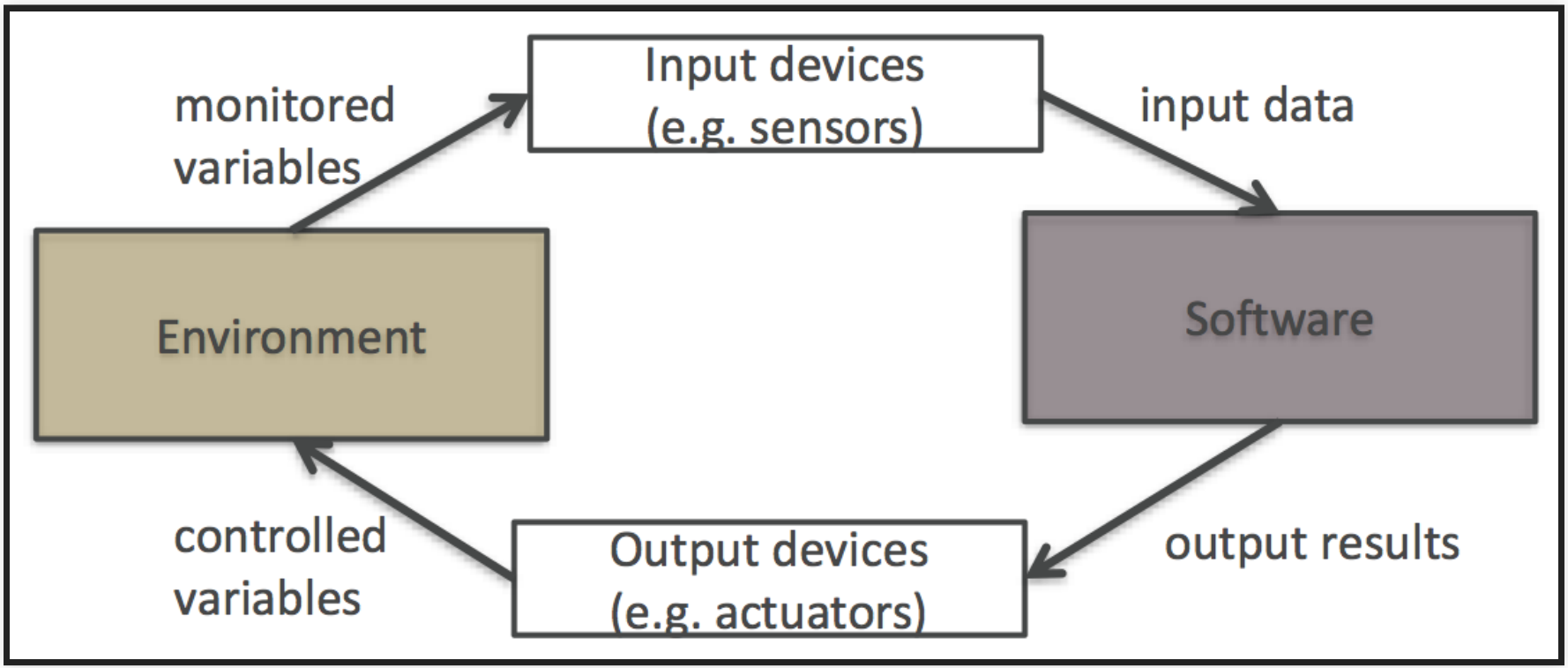
"The hardest single part of building a software system is deciding precisely what to build...No other part of the work so cripples the resulting system if done wrong." --
Fred Brooks, Mythical Man Month (1975)

IMPORTANCE OF REQUIREMENTS



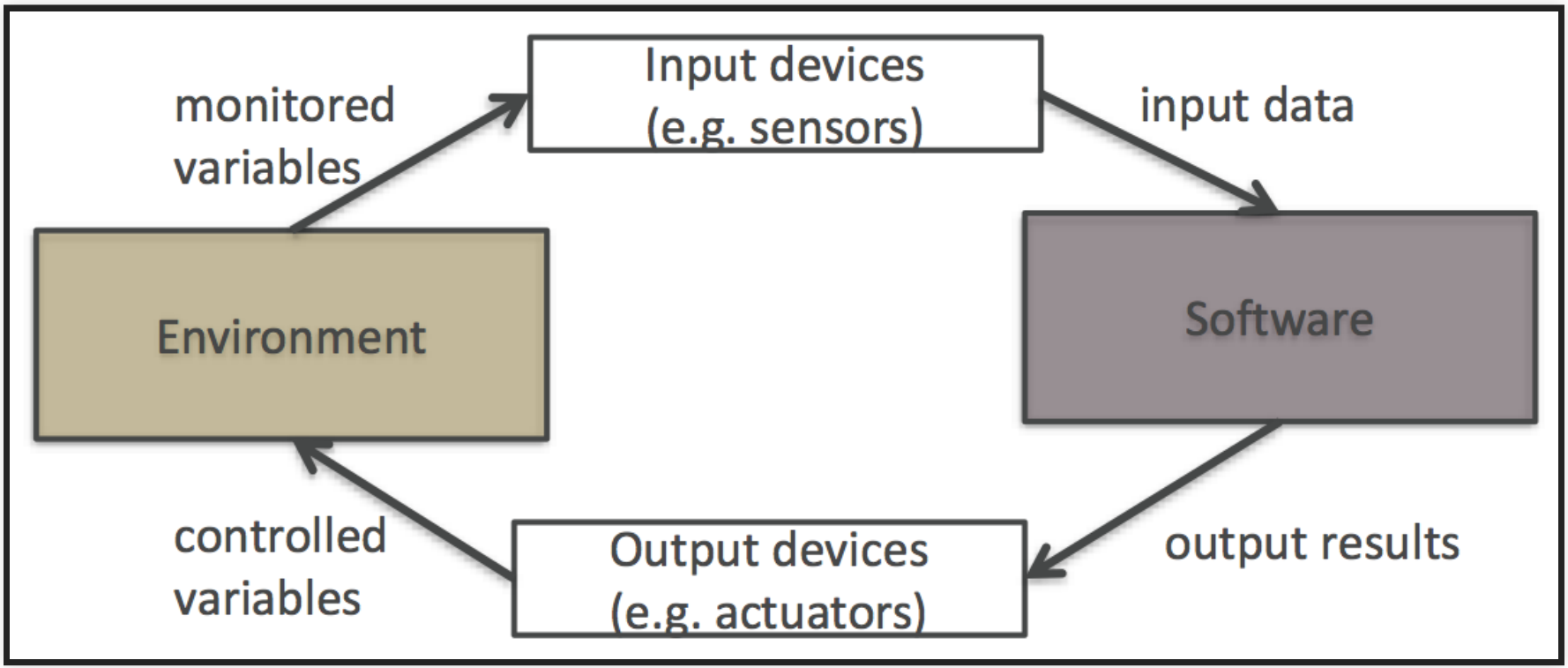
Only 3% of fatal software accidents due to coding errors; rest due to **poor requirements** or usability issues (National Research Council, 2007)

MACHINE VS WORLD



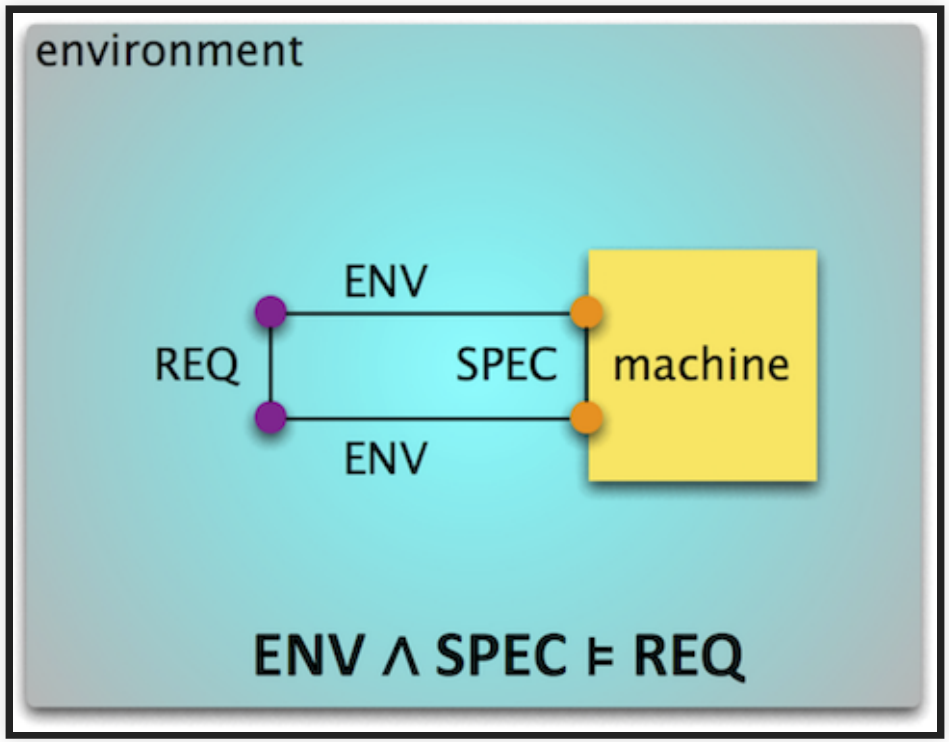
- No software lives in vacuum; every system is deployed as part of the world
- A requirement describes a desired state of the world (i.e., environment)
- Machine (software) is *created* to manipulate the environment into this state

MACHINE VS WORLD



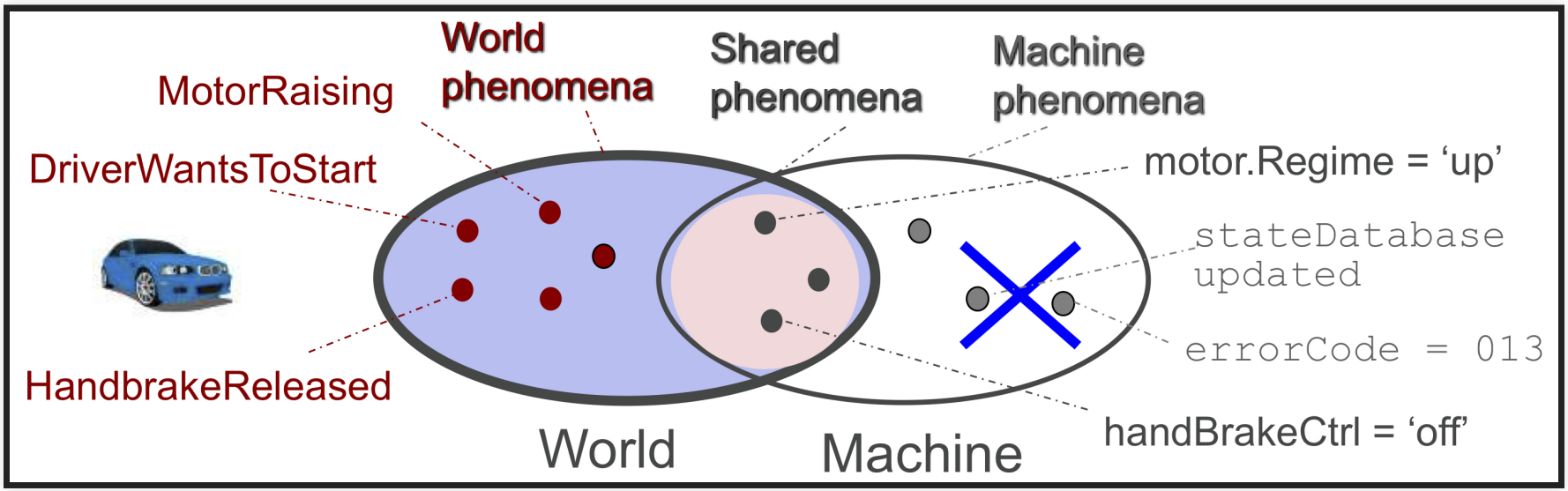
- Q. What is the environment for the following systems?
  - Self-driving car: ??
  - Smart home thermostats: ??
  - Movie recommender: ??

REQUIREMENT VS SPECIFICATION



- Requirement (REQ): What your product provides, as desired effects on the environment (i.e., system-level goals)
- Assumptions (ENV): What's assumed about the behavior/properties of the environment (based on domain knowledge)
- Specification (SPEC): What machine must do in order to satisfy REQ **in conjunction** with ENV

SHARED PHENOMENA



- Shared phenomena: Interface between the world & machine (actions, events, dataflow, etc.,)
- Requirements (REQ) are expressed only in terms of world phenomena
- Assumptions (ENV) are expressed in terms of world & shared phenomena
- Specifications (SPEC) are expressed in terms of machine & shared phenomena

EXAMPLE: LANE ASSIST

- Requirement (REQ): The vehicle must be prevented from veering off the lane.
- What are the entities in the environment?
- What about components in the machine?

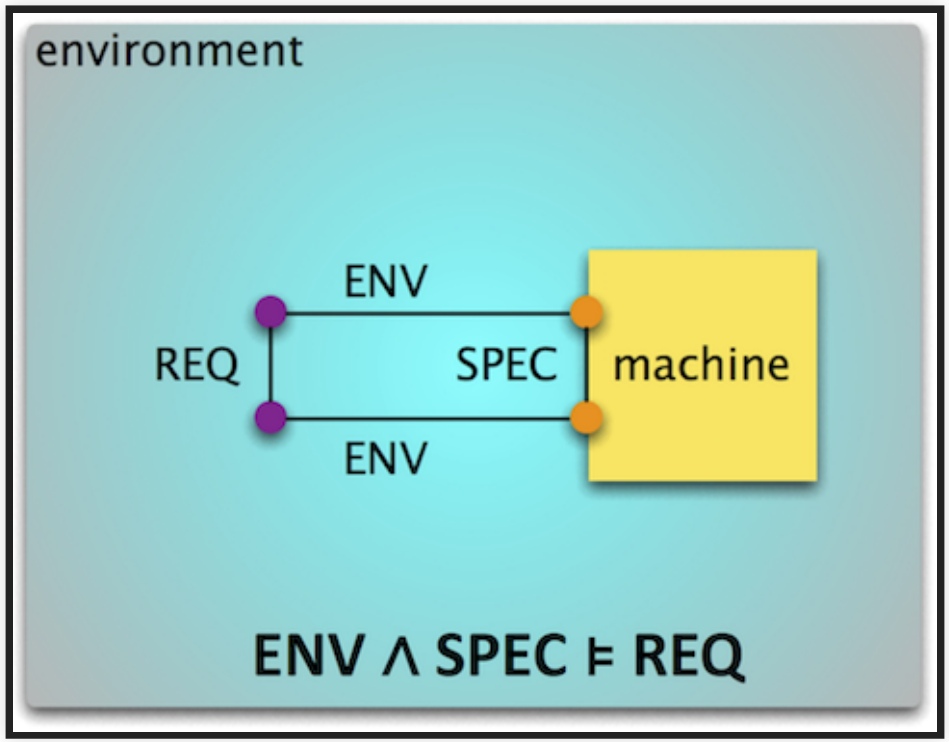
EXAMPLE: LANE ASSIST

- Requirement (REQ): The vehicle must be prevented from veering off the lane.
- Assumptions (ENV): ?
- Specification (SPEC): ?

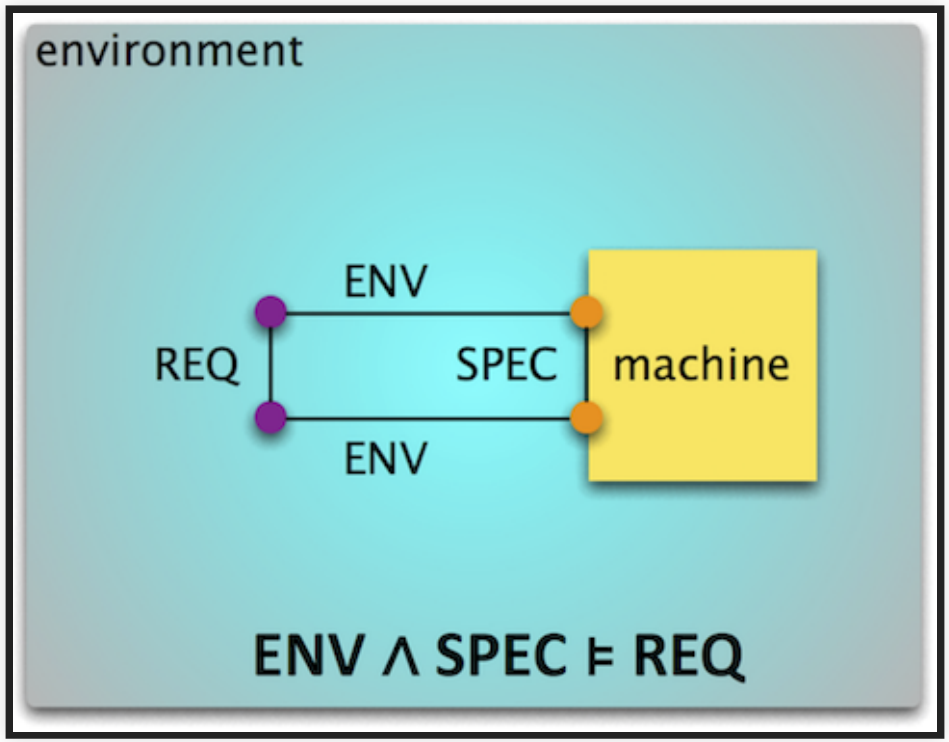
EXAMPLE: LANE ASSIST

- REQ: The vehicle must be prevented from veering off the lane.
- ENV: Sensors are providing accurate information about the lane; driver responses when given warning; steering wheel is functional
- SPEC: Lane detection accurately identifies the lane markings; the controller generates correct steering commands to keep the vehicle within lane

WHAT COULD GO WRONG?

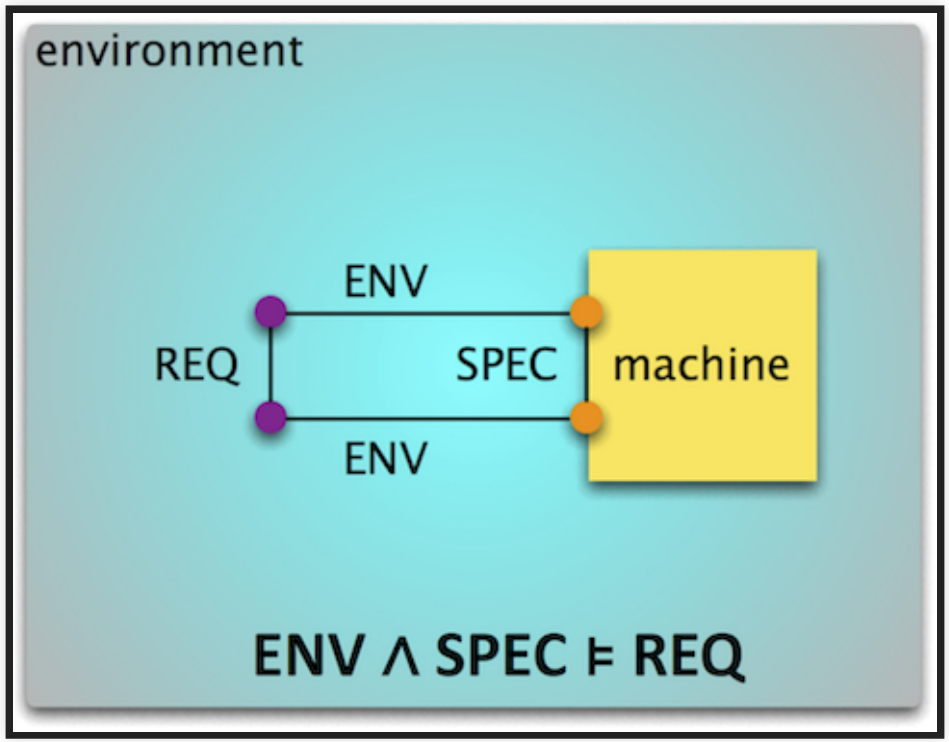


WHAT COULD GO WRONG?



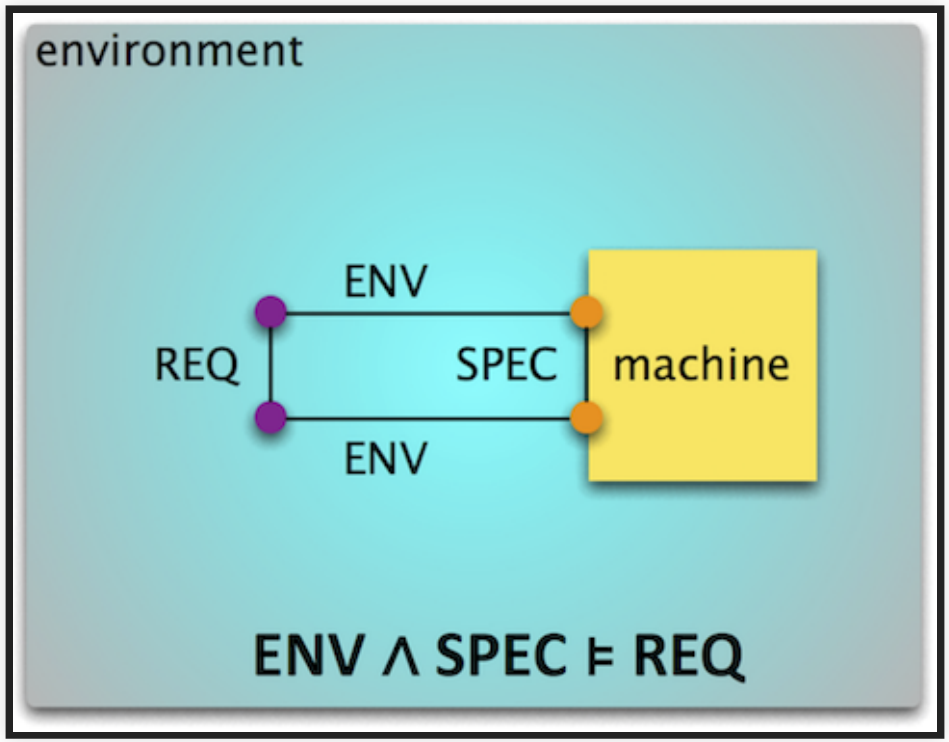
- Missing/incorrect environmental assumptions (ENV)

WHAT COULD GO WRONG?



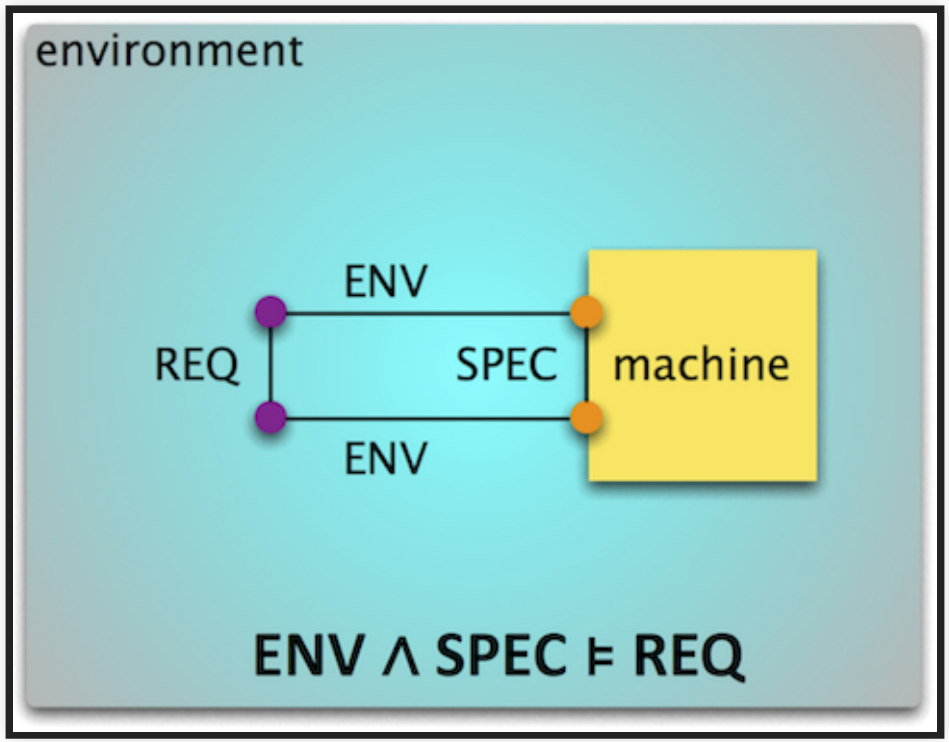
- Missing/incorrect environmental assumptions (ENV)
- Wrong/violated specification (SPEC)

WHAT COULD GO WRONG?



- Missing/incorrect environmental assumptions (ENV)
- Wrong/violated specification (SPEC)
- Inconsistency in assumptions & spec (ENV ∧ SPEC = False)

WHAT COULD GO WRONG?



- Missing/incorrect environmental assumptions (ENV)
- Wrong/violated specification (SPEC)
- Inconsistency in assumptions & spec (ENV ∧ SPEC = False)
- Inconsistency in requirements (REQ = False)

LUFTHANSA 2904 RUNAWAY CRASH

LUFTHANSA 2904 RUNAWAY CRASH

- Reverse thrust (RT): Decelerates plane during landing

LUFTHANSA 2904 RUNAWAY CRASH

- Reverse thrust (RT): Decelerates plane during landing
- What was required (REQ): RT enabled if and only if plane on the ground

LUFTHANSA 2904 RUNAWAY CRASH

- Reverse thrust (RT): Decelerates plane during landing
- What was required (REQ): RT enabled if and only if plane on the ground
- What was implemented (SPEC): RT enabled if and only if wheel turning

LUFTHANSA 2904 RUNAWAY CRASH

- Reverse thrust (RT): Decelerates plane during landing
- What was required (REQ): RT enabled if and only if plane on the ground
- What was implemented (SPEC): RT enabled if and only if wheel turning
- But runway wet due to rain
    - Wheel fails to turn, even though the plane is on the ground
    - Pilot attempts to enable RT; overridden by the software
    - Plane goes off the runway!

IMPLICATIONS ON SOFTWARE DEVELOPMENT

IMPLICATIONS ON SOFTWARE DEVELOPMENT

- Software/AI alone cannot establish system requirements
    - They are just one part of the system!

IMPLICATIONS ON SOFTWARE DEVELOPMENT

- Software/AI alone cannot establish system requirements
  - They are just one part of the system!
- Environmental assumptions are just as critical
  - But typically you can't modify these
  - Must design SPEC while treating ENV as given

IMPLICATIONS ON SOFTWARE DEVELOPMENT

- Software/AI alone cannot establish system requirements
  - They are just one part of the system!
- Environmental assumptions are just as critical
  - But typically you can't modify these
  - Must design SPEC while treating ENV as given
- If you ignore/misunderstand these, your system may fail to satisfy its requirements!

RECALL: LACK OF SPECIFICATIONS FOR AI COMPONENTS

- In addition to world vs machine challenges
- We do not have clear specifications for AI components (SPEC)
    - Goals, average accuracy
    - At best probabilistic specifications in some symbolic AI techniques
- Viewpoint: Machine learning techniques mine specifications from data, but not usually understandable
- But still important to articulate the responsibllities of AI components (SPEC) in establishing the system-level goals (REQ)

EXAMPLE: LANE ASSIST

- REQ: The vehicle must be prevented from veering off the lane.
- ENV: Sensors are providing accurate information about the lane; driver responses when given warning; steering wheel is functional
- SPEC: Lane detection accurately identifies the lane markings; the controller generates correct steering commands to keep the vehicle within lane

WHAT COULD GO WRONG IN LANE ASSIST?

- Missing/incorrect environmental assumptions (ENV)?
- Wrong/violated specification (SPEC)?
- Inconsistency in assumptions & spec (ENV ∧ SPEC = False)?
- Inconsistency in requirements (REQ = False)?

DERIVING SPEC FROM REQ

1. Identify environmental entities and machine components
2. State a desired requirement (REQ) over the environment
3. Identify the interface between the environment & machines
4. Identify the environmental assumptions (ENV)
5. Develop software specifications (SPEC) that are sufficient to establish REQ
6. Check whether $ENV \land SPEC \models REQ$
7. If NO, strengthen SPEC & repeat Step 6

Can't be automated! Domain knowledge is critical for coming up with REQ, ENV, and SPEC!

SUMMARY

- Accept that ML components will make mistakes
- Understand world-machine interactions
    - Machine vs World; specification vs requirements
    - Role of environmental assumptions in establishing requirements