Designing Systems for Security:

Walls are Only as Secure as They are Designed

Nigel Wright
October 18, 2022
2022 IUP CyberSecurity Day

IUP

Today’s Agenda

- Introduction
- System Design
  - What is a System
  - Allocation of System Features
- Design of Interfaces Elements
- Evaluating Tradeoffs
- Tools for Trade Off Consideration
- ABT - Always Be Testing
- Case Studies of Poor System Design

Hello!

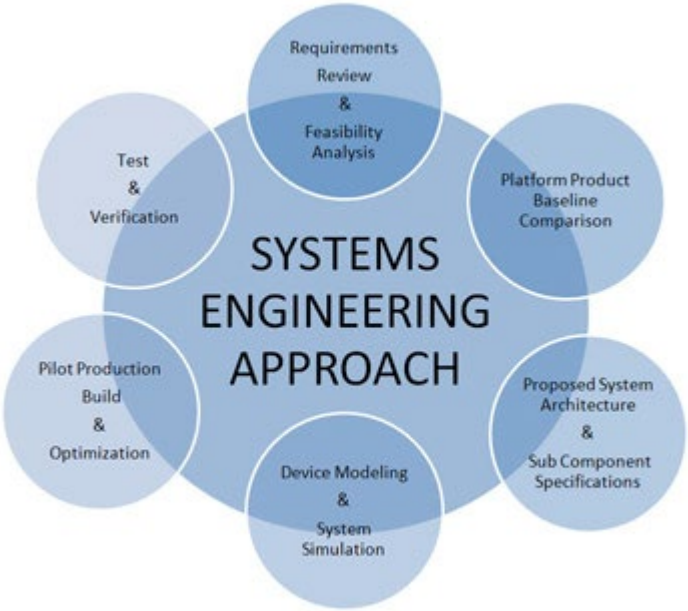
About Nigel!

- **Pittsburgher**
  Grew up south of Pittsburgh in California, PA.

- **Developing Safety Critical Autonomous Systems!**
  Defense, Medical Automated Rail Signaling Transit, RoboTaxi's and Human Lead Autonomous Trucks

Systems Design & CyberSecurity

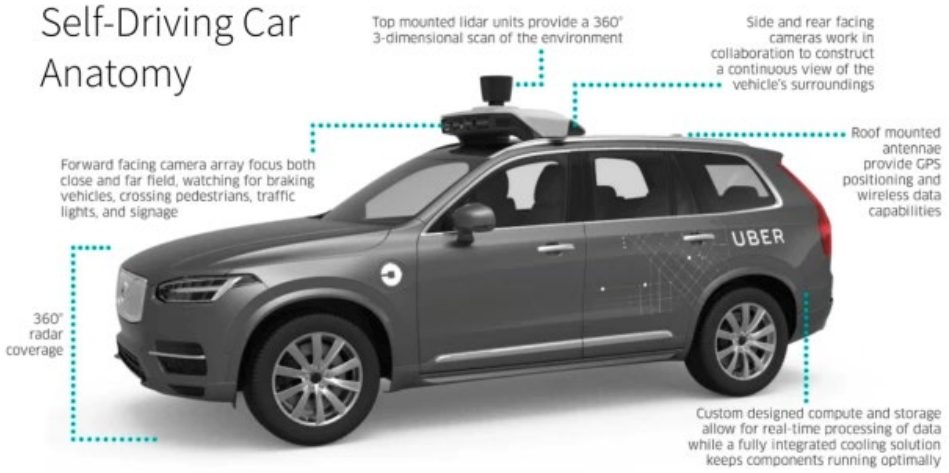
Approach to holistic look at development to ensure that the system (hardware and software) are free of vulnerabilities



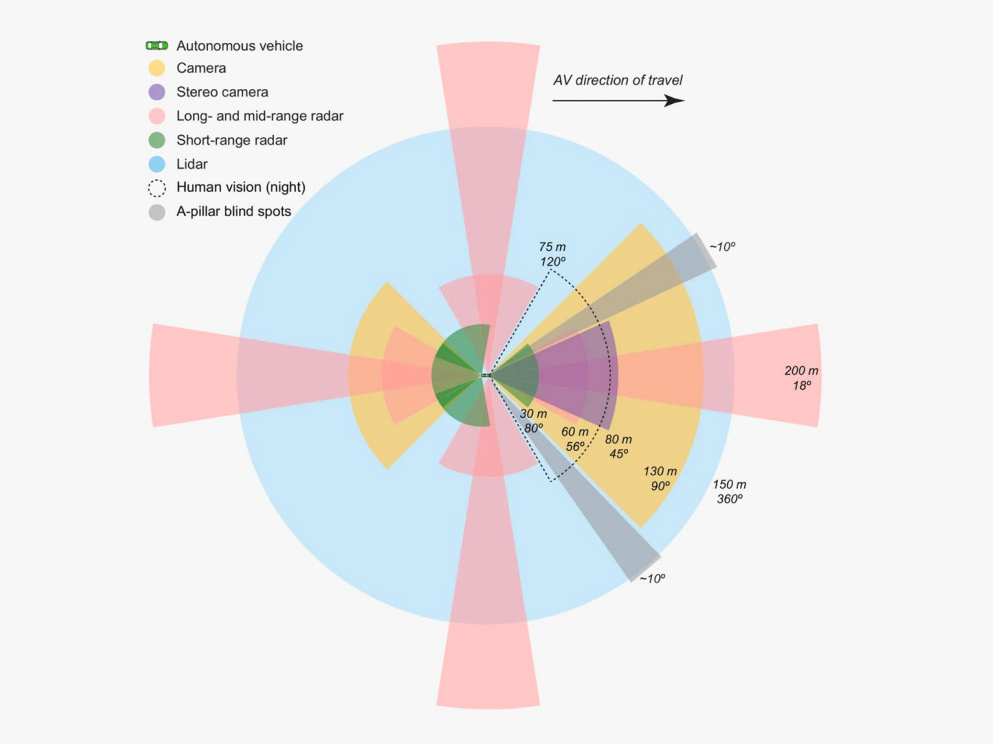
What is “Systems” Engineering

Systems engineering is an **interdisciplinary** field of engineering and engineering management that focuses on how to design and manage complex systems over their life cycles.

This includes, software, hardware, cyber-security, communications, supply chain…

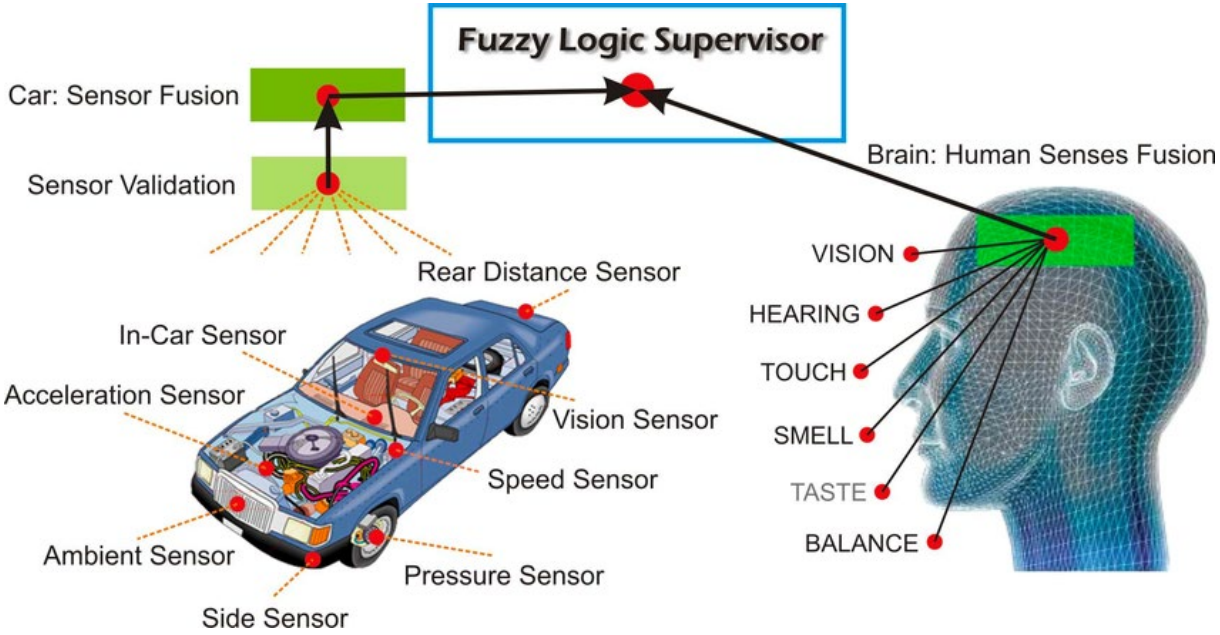


Applying “Systems” Understanding to specific contexts



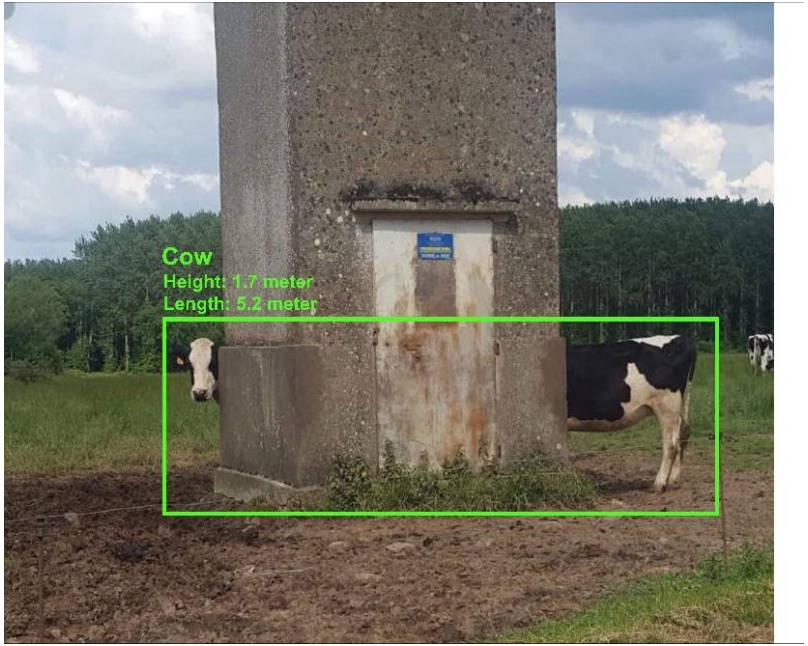
Example of a “generic autonomous vehicle sensor coverage overlay, color coded by sensor type.

Applying “Systems” Understanding to specific contexts



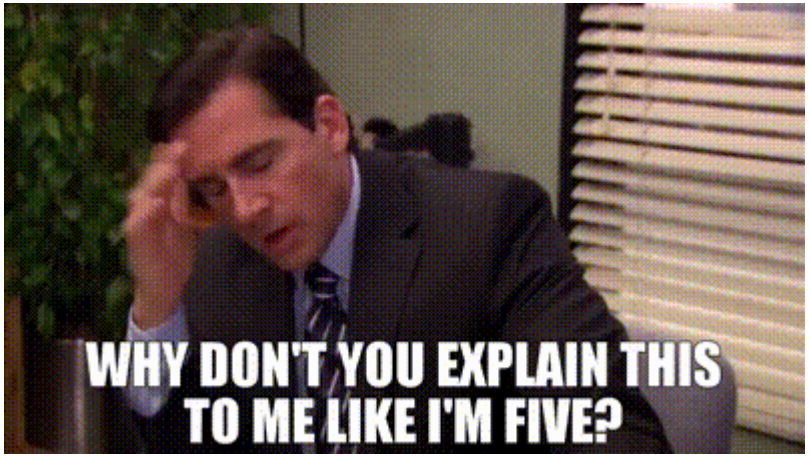
Sensing, and “computing” systems of a driving task

Applying “Systems” Understanding to specific contexts

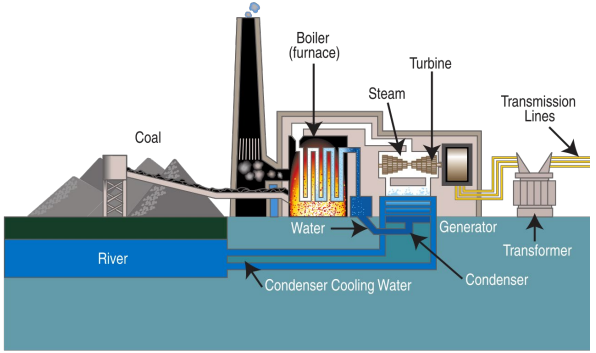


Example of a AI Detection

Using Systems Engineering to “Simplify” Understandings



Decomposing into a System Diagram

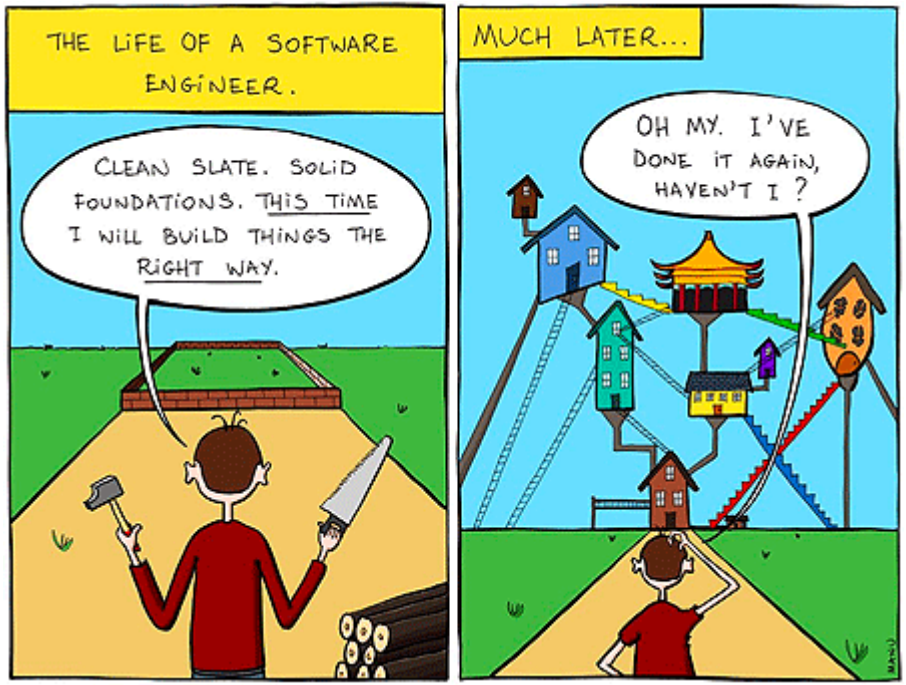


Systems Engineering methodologies enable architectural design

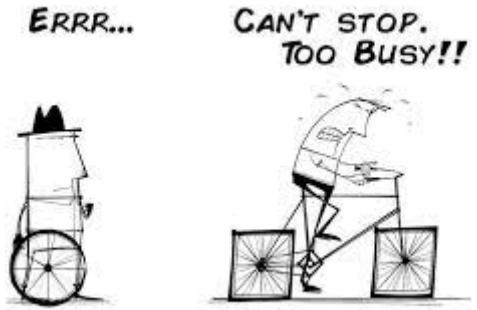
Systems Engineering methodologies enable designers to architectural decisions that support;

- security (data encapsulation)
- robustness (redundancy)
- abstraction (data exchange)

Systems Architecture



As engineers, you will need to be thoughtful about how the pieces of your systems interact…



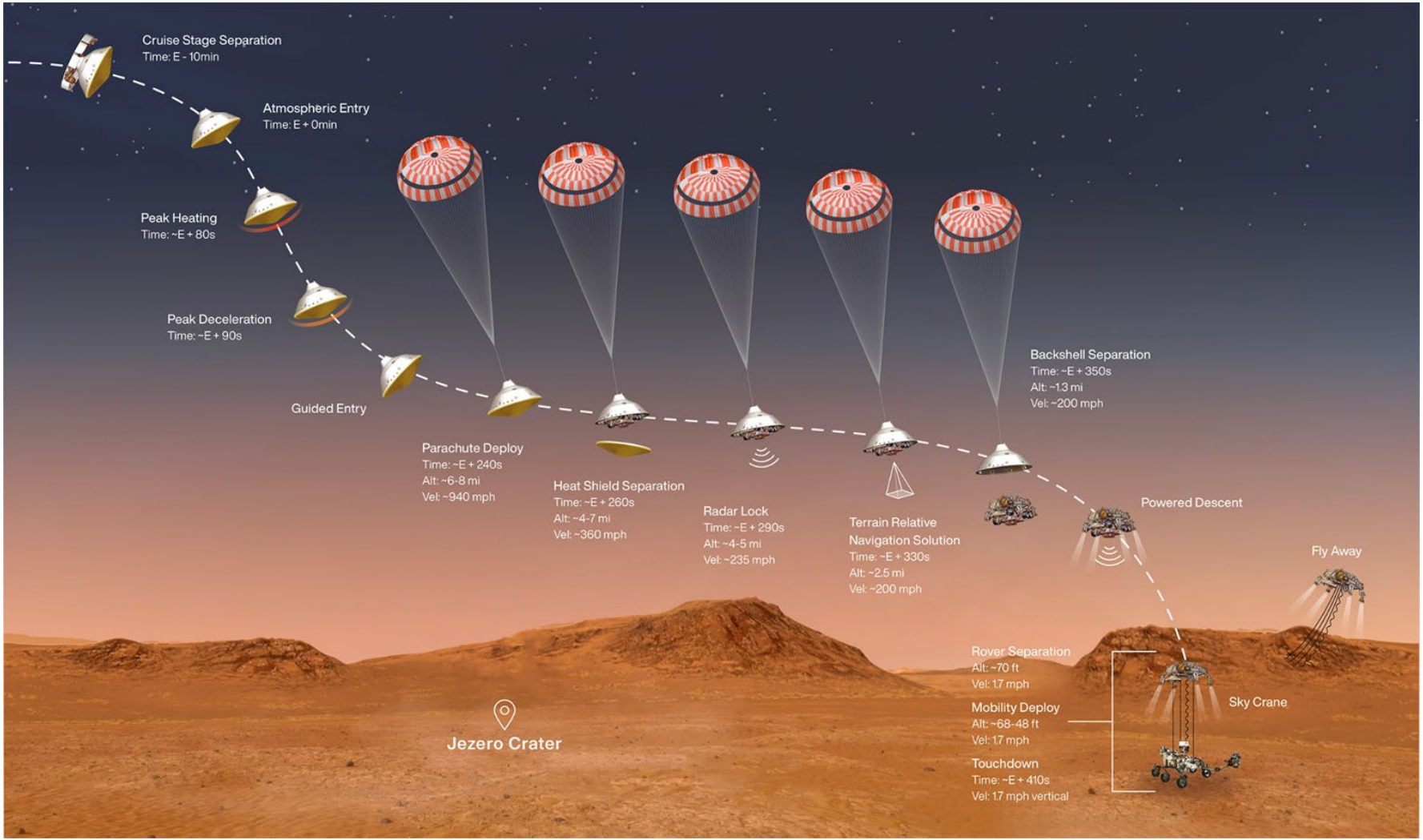
A “Secure” System

Everything is locked up tight!

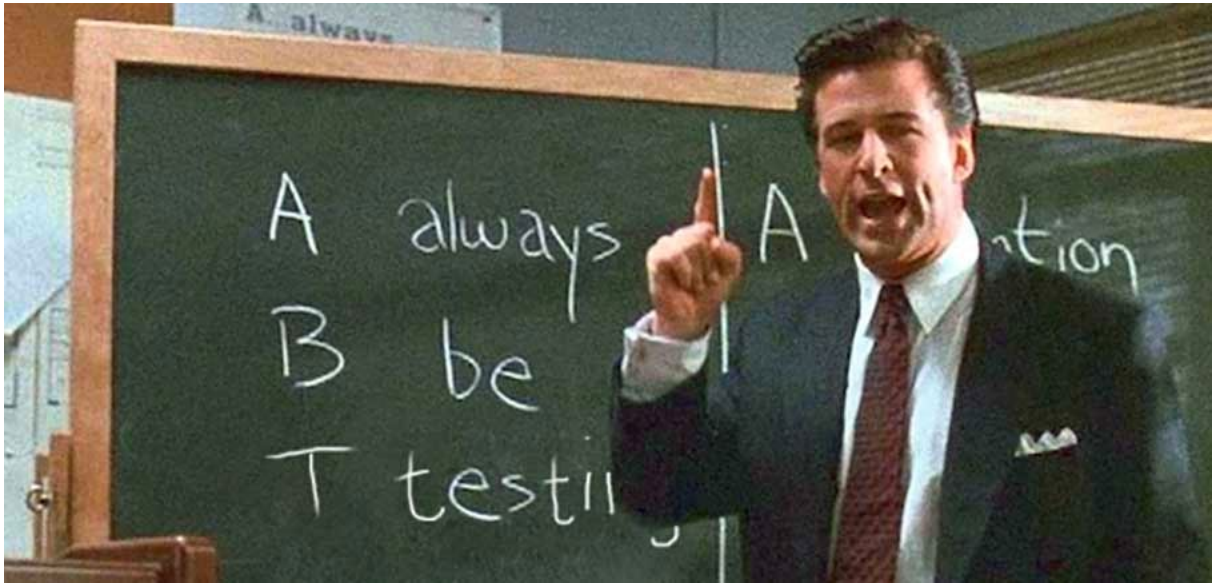
Toolkit - Concept of Operations

A “concept of operations” is a document describing the characteristics of a proposed system from the viewpoint of an individual who will use that system.

This enables all parties to understand at a glance the way that a complex system is intended to behave.

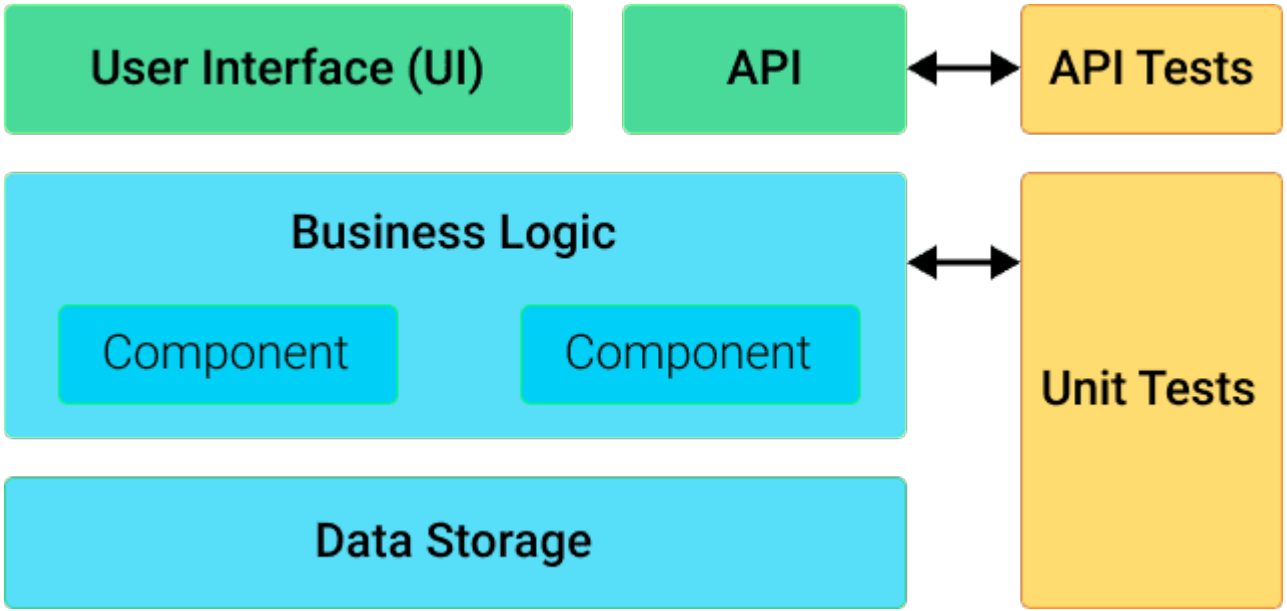

Cruise Stage Separation
Time: E - 10min
Atmospheric Entry
Time: E + 0min
Peak Heating
Time: ~E + 80s
Peak Deceleration
Time: ~E + 90s
Guided Entry
Parachute Deploy
Time: ~E + 240s
Alt: ~6-8 mi
Vel: ~940 mph
Heat Shield Separation
Time: ~E + 260s
Alt: ~4-7 mi
Vel: ~360 mph
Radar Lock
Time: ~E + 290s
Alt: ~4-5 mi
Vel: ~235 mph
Terrain Relative Navigation Solution
Time: ~E + 330s
Alt: ~2.5 mi
Vel: ~200 mph
Backshell Separation
Time: ~E + 350s
Alt: ~1.3 mi
Vel: ~200 mph
Powered Descent
Fly Away
Rover Separation
Alt: ~70 ft
Vel: 17 mph
Mobility Deploy
Alt: ~68-48 ft
Vel: 17 mph
Touchdown
Time: ~E + 410s
Vel: 17 mph vertical
Sky Crane
Jezero Crater

ABT - Always Be Testing



Glenn Gary Glenn Ross's famous scene of system validation

ABT - Always Be Testing - Examples



System Diagram of a Software Application

ABT - Always Be Testing - Examples

Function Testing

Bounds Checking for return values

Range Sweeping for incoming parameters

What happens if this function is passed a null pointer, an invalid parameter?

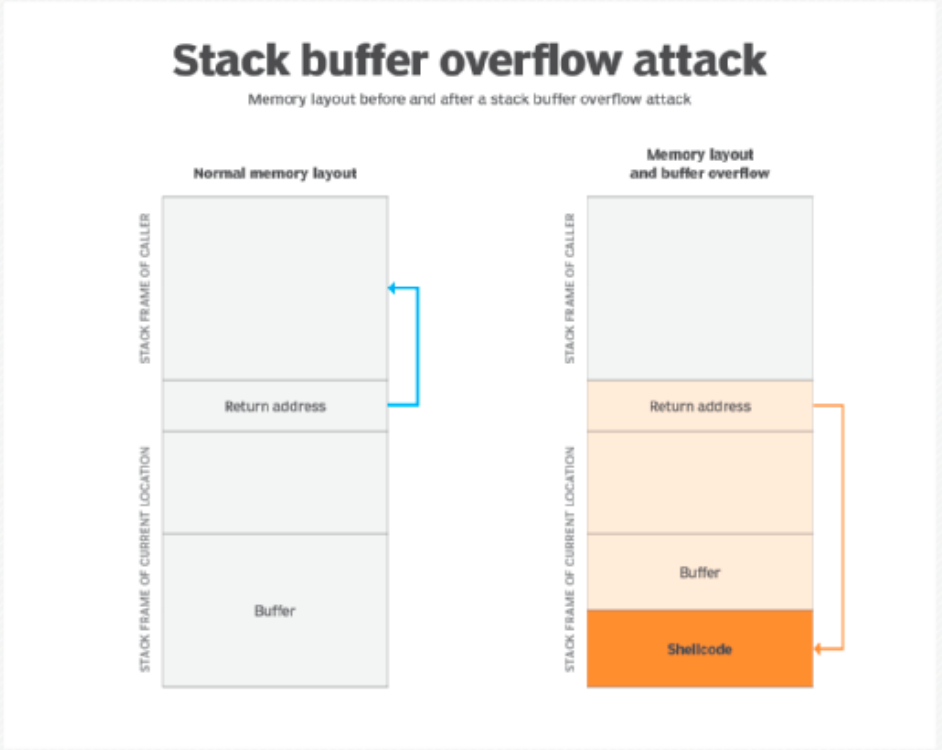
Are incoming parameters sanitized?

Can this function be overloaded?

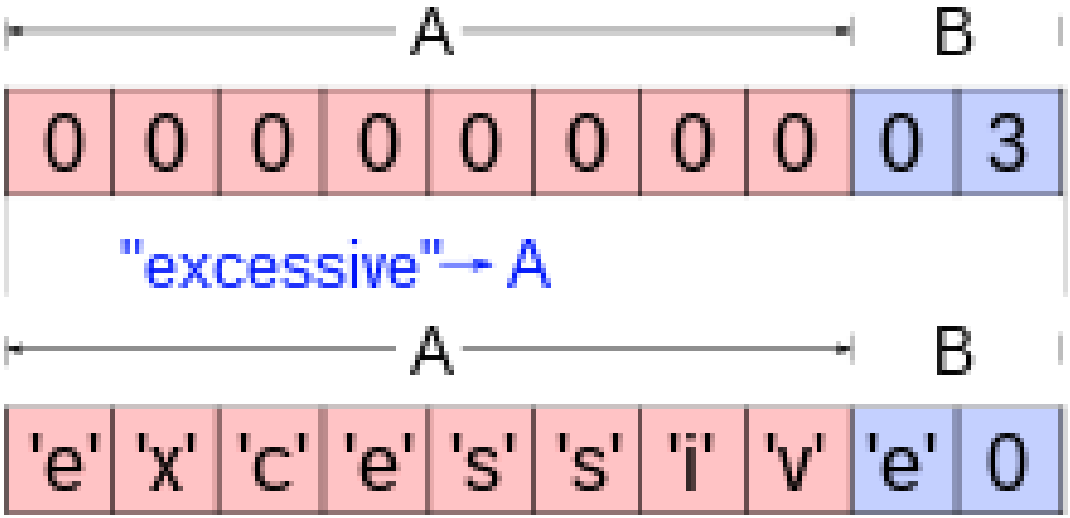
Buffer Overflow Examples

A buffer overflow occurs when a program is able to write more data to a buffer—or fixed-length block of computer memory—than it is designed to hold.

Then the excess data will overflow into the adjacent buffer, overwriting its contents and enabling the attacker to change the flow of the program and execute a code injection attack.

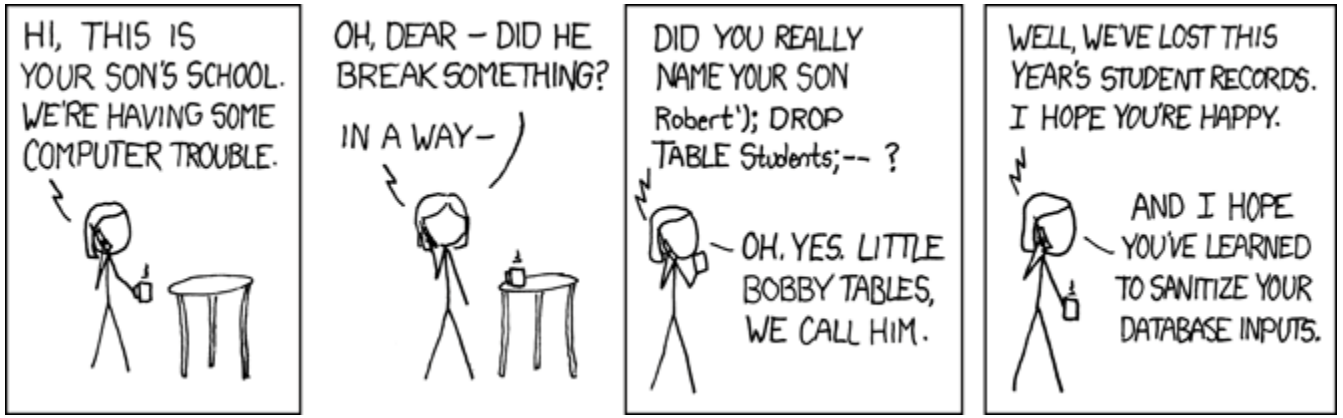


Buffer Overflow Examples

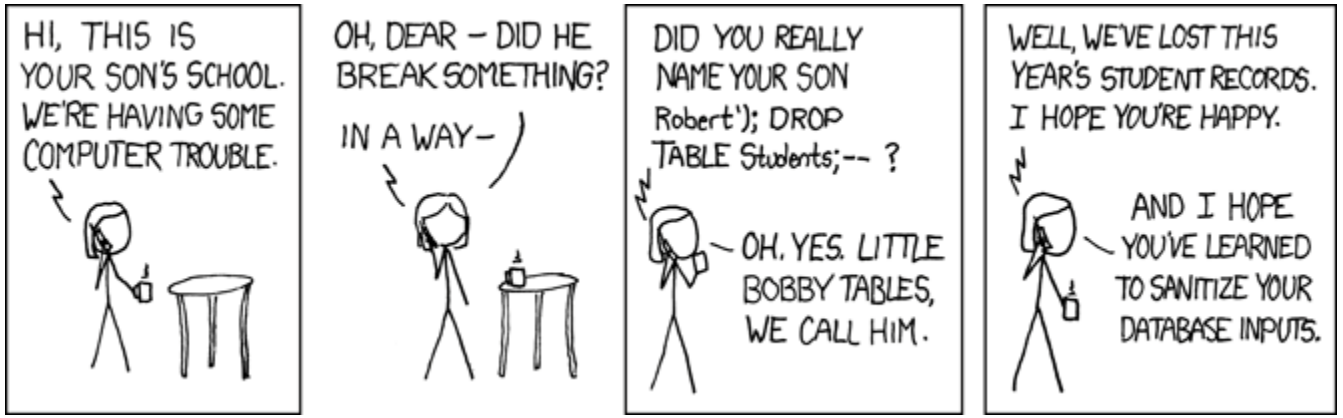


String “excessive” stepping into variable space for B

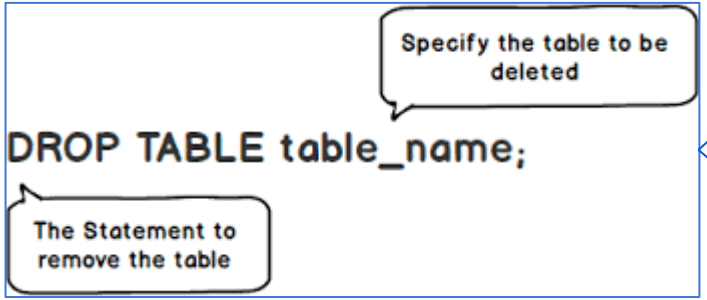
Example Sanitizing Inputs: Little Johnny Drop Tables

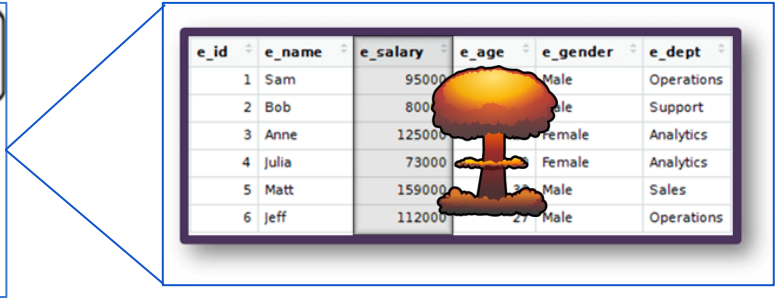


Example Sanitizing Inputs: Little Johnny Drop Tables



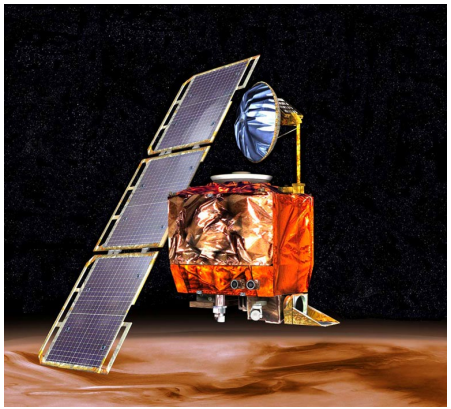
SQL Primer;

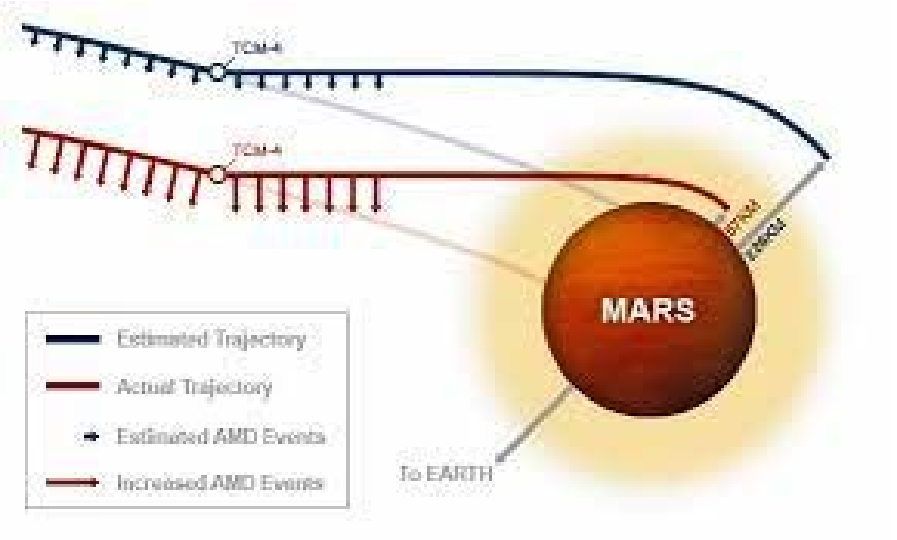




Integration Errors - Input Type Checking

1999, Nasa burnt up a $200 million dollar Mars Climate Orbiter when the engineers failed to realize one function was expecting units in **english**, the other in **metric…**





NASA Climate Orbiter

Integration Errors - Input Type Checking

Remember the Mars Climate Orbiter incident from 1999?

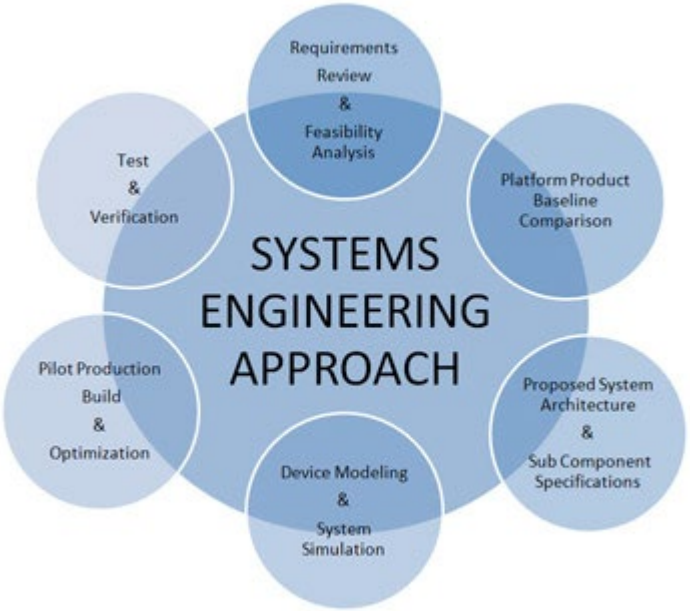
Closing Thoughts. . .

Ensure that you take a step back and think of the overall “system” and it’s use…

Closing Thoughts. . .

Closing Thoughts. . .

Taking a structured, pragmatic approach enables rapid deployment… and ultimately enables a more robust, scalable and safe system…



Q&A