Week 8

ENGR489
Engineering Design

Alvin C. Valera

alvin.valera@ecs.vuw.ac.nz

learning
system
computer
programming
vehicle
game
project
analysis
network
data
proposal
video
honeypot
experience
energy
collecting
software
machine
counterfactual
interface
mobile
monitoring
explainable
collection
automated
dc
editor
tool
using
robot
panel
creating
open
solar
science
test
rig
image
uav
detection
based
intelligence
decentralised
iot
cloud
public
community
approach
teaching
rat
dynamic
electric
performance
automating
humans
services
control
tracking
genetic
mussel
improved
user
source
walk
deep
characterisation
ai
streaming
mechatronic
recognition
quality
diabetes
leader
outreach
parsing
buck
caching
turk
mechanism
implementation
applications
distinguish
stick
strategies
converter
pipelines
household
bots
object
switch
information
big
car
shell
harvesting
storage
stereo
mapping
teach
prevention
course
app
named
decision
wild
exchange
making
vocalisations
brokerless
tagging
expressive
connection
exposure
thermal
emotional
databases
ui
platform
networking
vulnerability
habitats
typing
breeding
melon
animals
learn
visual
figures
lightweight
waste
sickness
wildlife
text
noising
communication
speed
grid
inpainting
improve
understanding
follower
health
unsupervised
visualisation
determining
fingerprinting
de
renewable
access
static
wiring
improving
air
fish
bytecode
bait
generated
cable
ultrasonic
concepts
techniques
tempering
better
eight360
effective
iow
xr
simulators
chordophone
subscribe
cost
transport
nanogrid
motion
handover
fighting
visualising
locally
motor
ga
vr
bifacial
thermoelectric
internet
optimising
cowrie
designing
multibot
patterns
robotic
formation
temple
publish
ux
social
ecs
rf
graph
establishing
sword

Content

- **A look back: What is Engineering?**

- **Engineering Design**

- **What Next?**

Main source: SWEBOK v3.0

What is engineering?

Image source: https://gradaustralia.com.au/career-planning/which-engineering-specialisation-is-right-for-me-here-are-a-few

- "the application of *science* and *mathematics* by which the properties of matter and the sources of energy in nature are made useful to people" [Merriam-Webster]

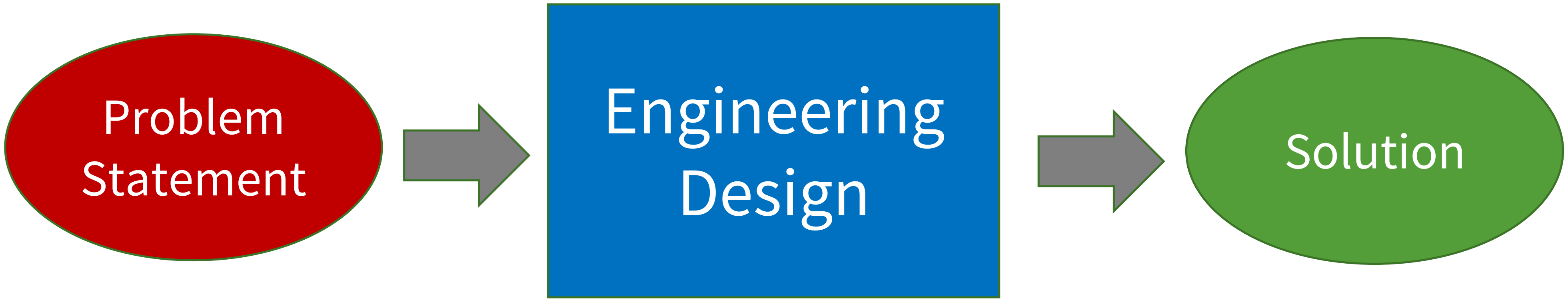
What is engineering?

Image source: https://gradaustralia.com.au/career-planning/which-engineering-specialisation-is-right-for-me-here-are-a-few

- “the application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems or processes” [IEEE]

Engineering design

- Engineering design is the process of devising a solution to a given problem through a systematic, disciplined, quantifiable approach



- Engineering design is a problem solving activity to come up with a feasible solution from a set of possible solutions

Feasible solution

- In engineering, a problem usually has many possible solutions that satisfy requirements

- Engineering design involves choosing a feasible solution that satisfy constraints
  - Example constraints: cost, power source, physical dimensions or weight

“Wicked problem”

- Many engineering problems can be considered “wicked problems”
    - **Open ended and vaguely defined**
    - There are usually several alternative ways to solve the problem

- How to solve wicked problems?

Source: https://www.wicked7.org/what-is-a-wicked-problem/

Solving a wicked problem

- A wicked problem is one that could be clearly defined only by solving it or by solving part of it

- **Therefore: A wicked problem has to be solved once in order to define it clearly and then solved again to create a solution that works**

Steps in engineering design

1. Define the problem
2. Gather pertinent information
3. Generate multiple solutions
4. Analyse and select a solution
5. Implement the solution

- **Not necessarily linear but more iterative: knowledge gained at any step may be used to inform earlier tasks and an iteration in the process**

Defining the problem

[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

- **Done at the proposal stage**

- Involves gathering the requirements
  - For industry projects, this may involve identifying product functions and features

- Refining the problem statement to identify the real problem to be solved and setting the design goals
  - For certain projects (e.g. industry projects), this may also involve stating the project *success criteria*

Gathering pertinent information

- Expand your knowledge about the problem and existing solutions
- Conduct background research to know how existing work solves the problem and their limitations
- This phase may reveal facts that can lead to the redefinition of the problem

Generating multiple solutions

- Conceptualize multiple possible solutions and refine them to a **sufficient level** of detail that a comparison can be done among them

▸ Articulating your design

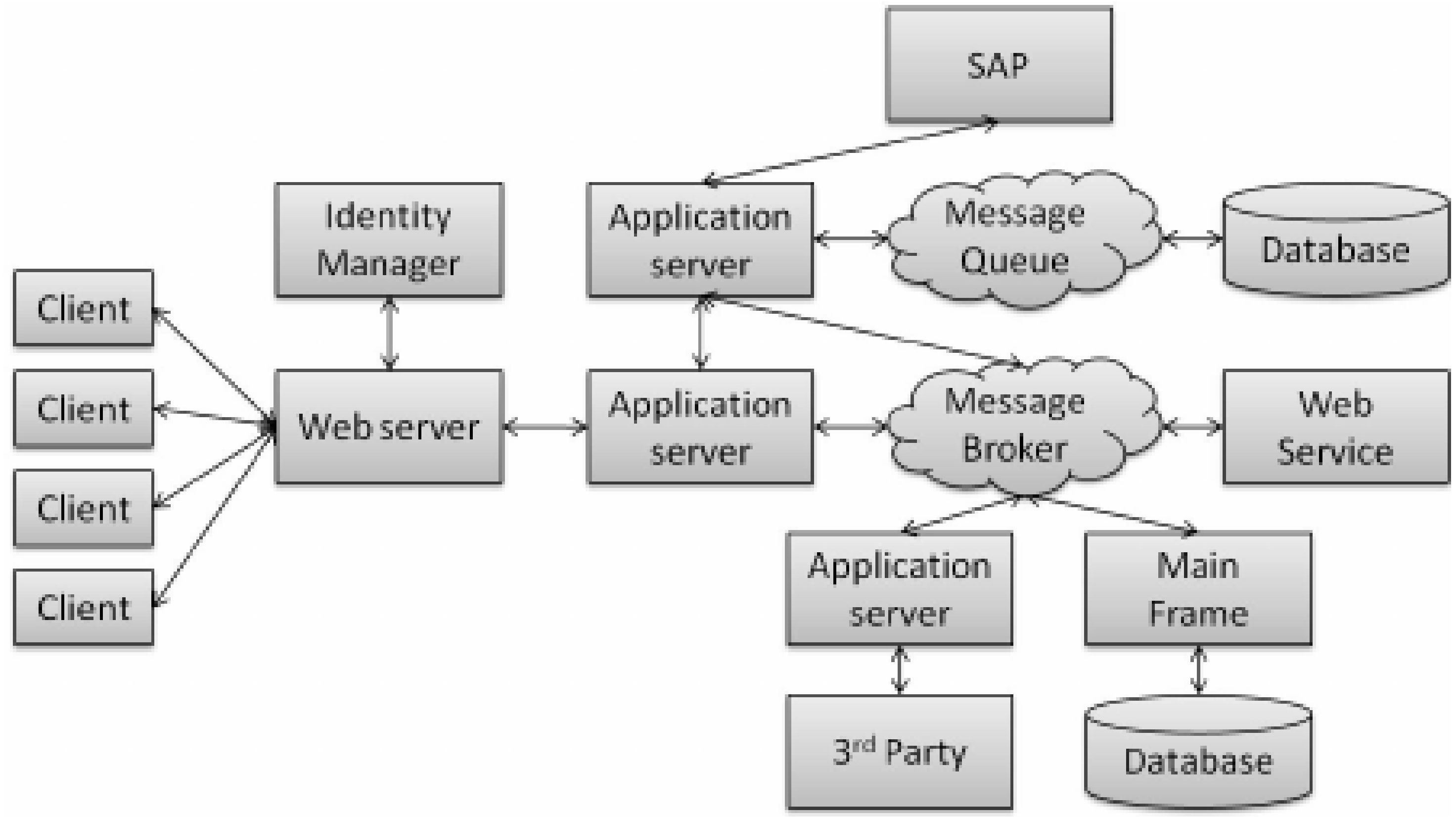
- Articulating and documenting your design solution is critical as often the case, someone else would be implementing it

- This usually involves the use of appropriate diagrams with different level of detail:
  - **High-level design**: usually refers to the system architecture and is depicted in a systems architecture diagram
  - **Detailed or low-level design**: usually refers to the design of every component in the system

▶▶ High-level design

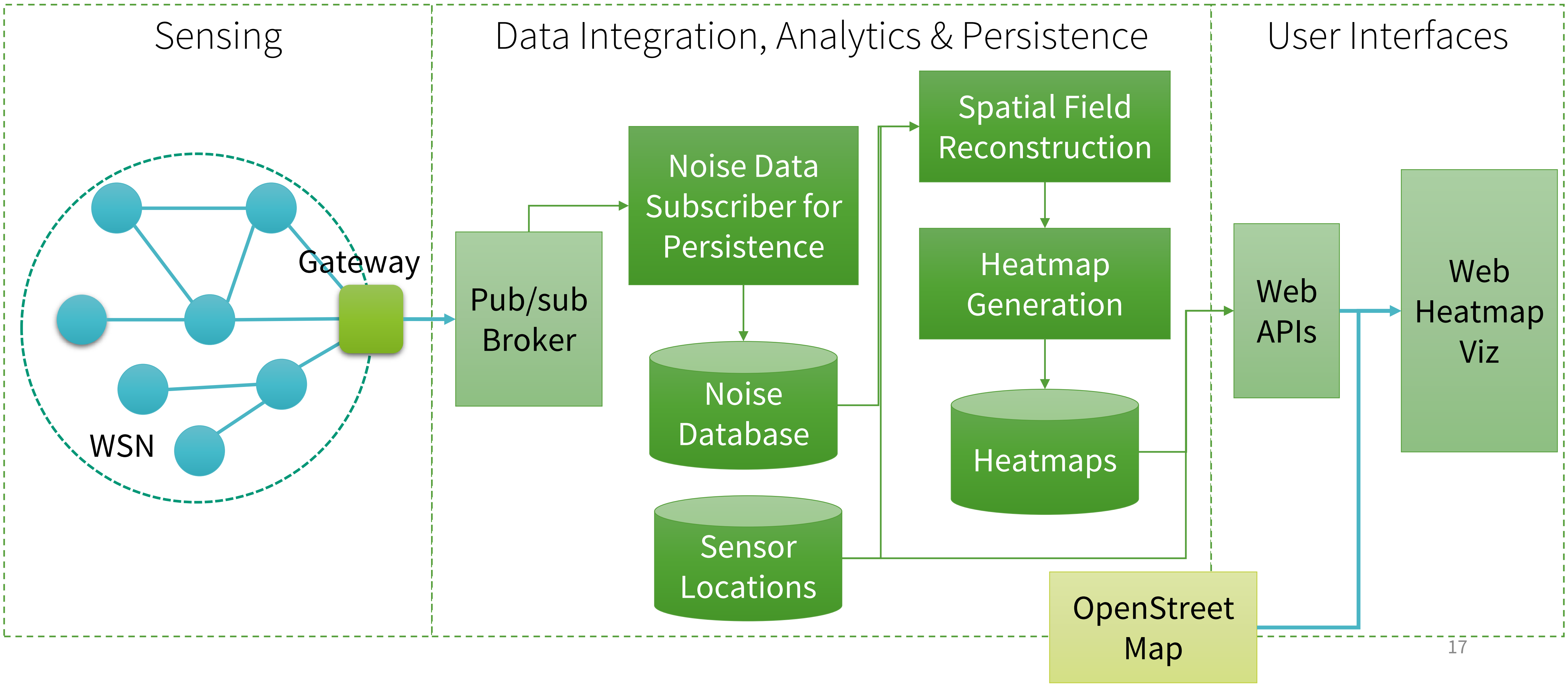
- Use **systems level** diagrams that you have learnt in the past
- Usually includes the different systems/subsystems that your project is using/interfacing with
- Show how your system/subsystem interacts with the other components, including external stakeholders
  - Interacting components are usually directly connected using lines or arrows
- **At the systems level, defer decisions as much as possible!**

▶▶ System architecture example 1



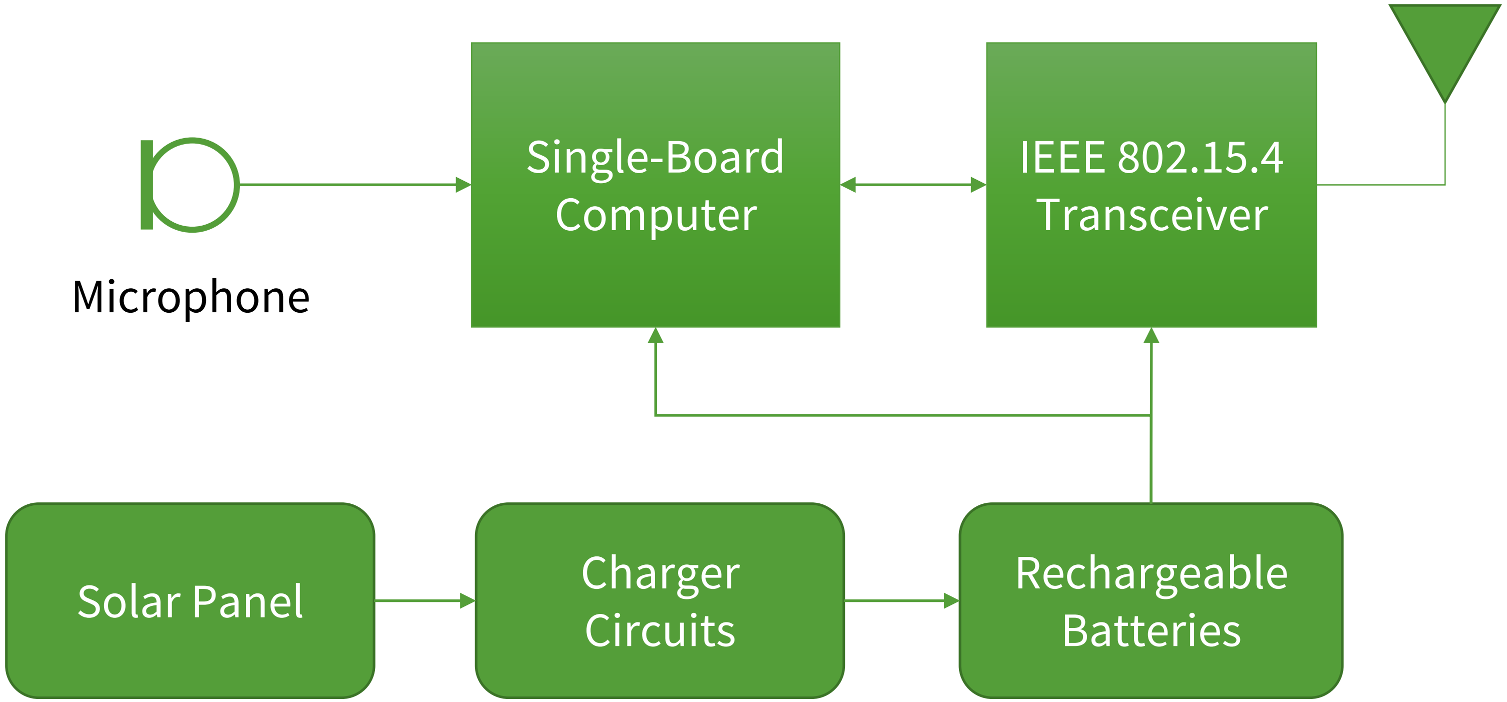
Source: Rabl, Tilmann, Hans-Arno Jacobsen, and Serge Mankovskii. "Big data challenges in application performance management." *Proc. 5th Extremely Large Database Conf.*. 2011.

▶▶ System architecture example 2



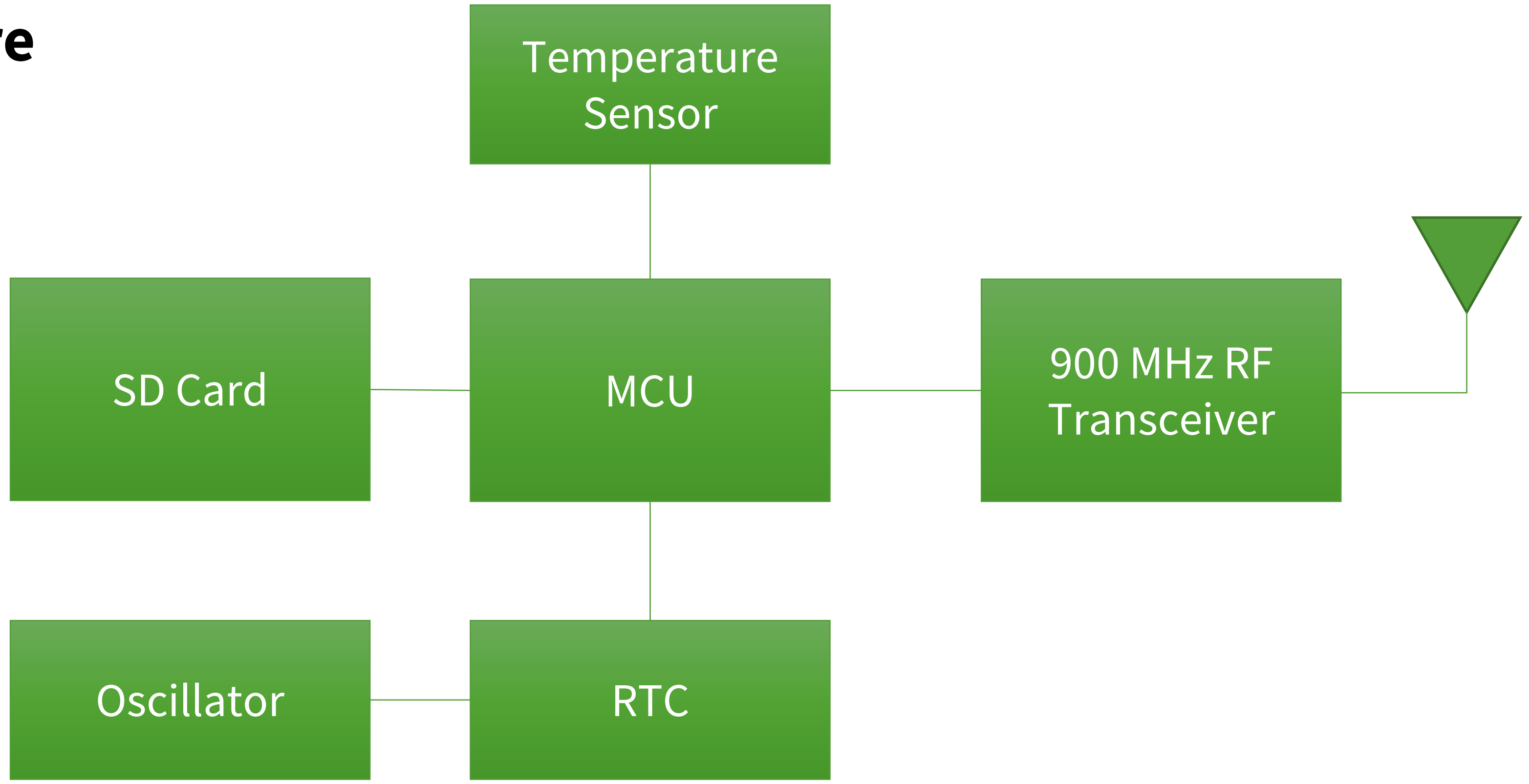
▶▶ System architecture example 3

Hardware



▶▶ System architecture example 4

Hardware



Source: Dinesh, M., and K. B. Bhaskar. "Smart Highway Accident Alert Using Raspberry Pi Camera." *Journal of Digital Integrated Circuits in Electrical Devices* 5 (2020).

▶▶ Low-level design

- Use appropriate diagrams that you have learnt in the past

- For software, you can use the 4+1 architectural view model

- For algorithms and protocols, flowcharts, state diagrams, message sequence diagrams, timing diagrams or even pseudocodes would also be suitable

- For hardware, you can use circuit schematics diagrams and/or wiring diagrams

Analysing and selecting a solution

- **Functional analysis** to assess whether proposed design would meet the functional requirements
- For designs involving human users, there is also a need to analyse **ergonomics** and **user-friendliness**
- Other aspects include: **performance**, **cost**, **safety**

- The types and amount of analysis depends on type of problem and the needs that the solution must address, and the constraints imposed on the design

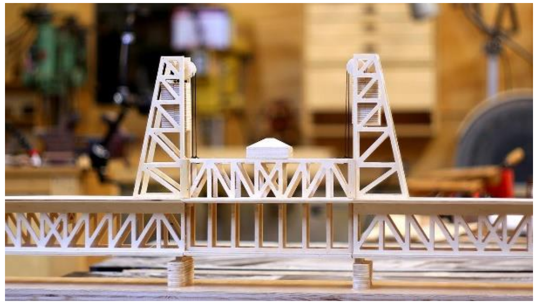
▶ Modelling & simulation

- **Modelling**: abstraction process used to represent some aspects of a system
- **Simulation**: uses the model and provides a means of conducting designed experiments with that model to better understand the system, its behaviour, and relationships between subsystems, as well as to analyse aspects of the design

- Engineers use modelling and simulation to construct theories or hypotheses about the behaviour of the system, then use those theories to make predictions about the system

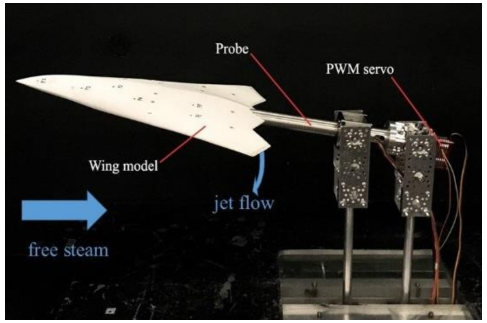
▸ Model types

- **Iconic:** *visually equivalent* but incomplete 2-dimensional or 3-dimensional representation



Source: 1

- **Analogic:** *functionally equivalent* but incomplete representation; the model behaves like the physical artefact even though it may not physically resemble it



Source: 2

- **Symbolic:** model is represented using symbols such as equations and captures relevant aspects of the process or system in symbolic form

$$F = ma$$

1) https://www.popularmechanics.com/technology/infrastructure/a20907/third-grader-bridge/

2) He, Xiaowei, Mathieu Le Provost, and David R. Williams. "Dynamic active flow control of the roll moment on a generic ucas wing." *2018 AIAA Aerospace Sciences Meeting*. 2018.

Implementing the chosen solution

- Development and testing of the chosen design
- May involve the development of a **prototype** to test the chosen solution under certain conditions
  - Feedback may be used to refine design or motivate the selection of an alternative design solution
- **Documentation of the design solution as well as of the tradeoffs for the choices made in the design of the solution is critical!**

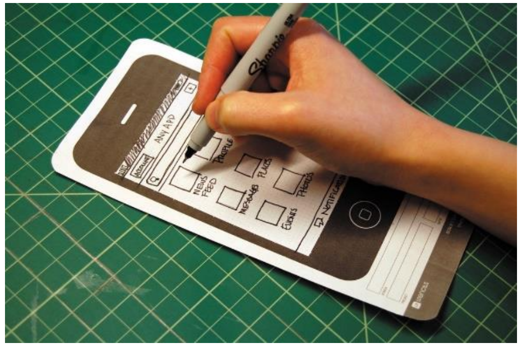
▸ Prototyping

- An abstraction process where a partial representation (that captures aspects of interest) of the product or system is built
- A prototype may be an initial version of the system but lacks the full functionality of the final version
- For hardware, the prototype may actually be the first fully functional version of a system or it may be a model of the system
- For software, the prototype is an abstract model of part of the software, not constructed with all of the architectural, performance, and other quality characteristics expected in the finished product

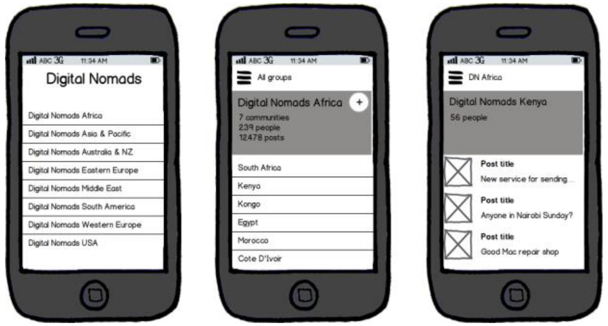
▸ Software prototypes (4 common types)

- Source: https://medium.com/fold-line-gold/four-common-types-of-software-prototypes-8fa275c0602f

Low fidelity



Middle fidelity



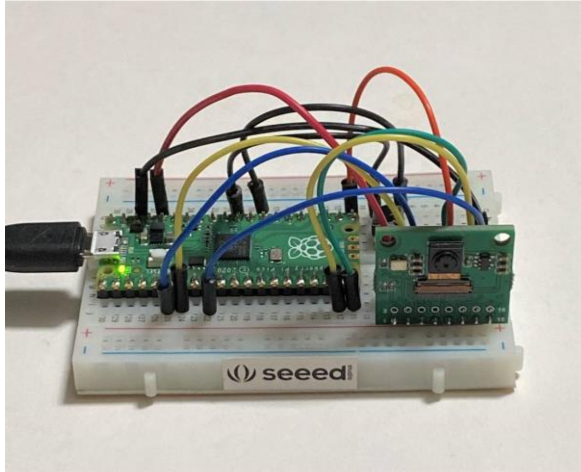
High fidelity

Super fidelity

▶ Hardware prototypes (3 stages)

- Source: https://predictabledesigns.com/the-essential-guide-to-prototyping-your-new-electronic-hardware-product/

Proof of concept (POC)

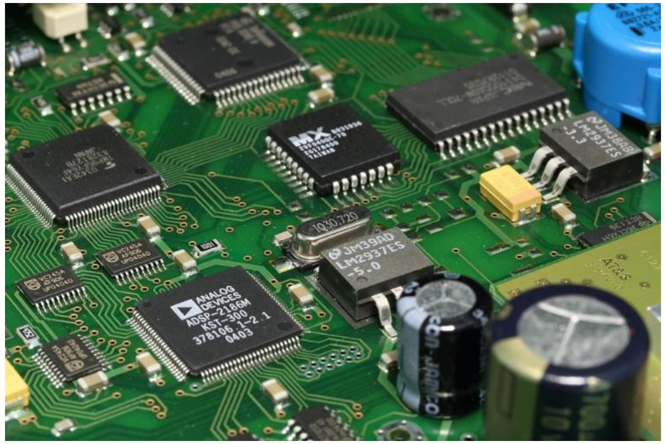


Source: https://community.element14.com/members-area/personalblogs/b/ralph-yamamoto-s-blog/posts/camera-module-for-raspberry-pi-pico

“Looks like” prototype

Source: https://medium.com/abilista/prototype-of-your-invention-b87e8efb0c08

“Works like” prototype



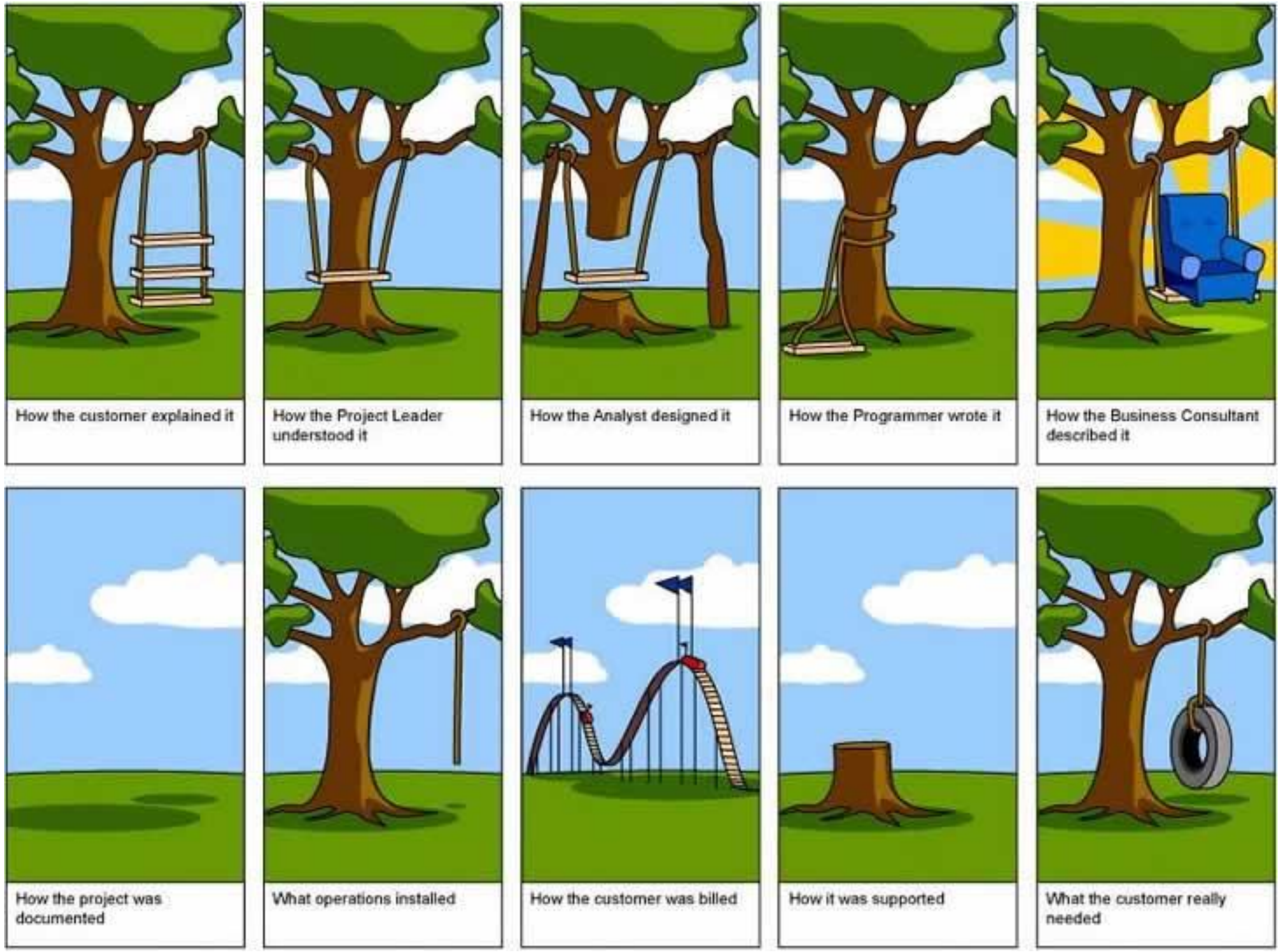
Source: https://predictabledesigns.com/the-essential-guide-to-prototyping-your-new-electronic-hardware-product/

Reality check

- Your design probably ignored unknown unknowns
  - System may behave unexpectedly in certain situations
- Your system has higher probability of failing than working…
  - Design errors
  - Implementation errors
  - Runtime errors
  - Many other sources of noise/error
- How to improve success probability?
  - Test thoroughly to uncover many errors
  - Design for robustness: make system work correctly even in the presence of errors!

And even when it works most of the time…

Does it solve the problem that you want to solve in the first place?


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

What next? Design review

- You should engage with your supervisor or peers for feedback before prototyping/implementing your chosen solution

- Suggested activity:
  1. Develop several design solutions to your problem
  2. Choose one of the design solutions using appropriate analysis
  3. Document your decisions and chosen design using appropriate diagrams and explanations
  4. Present your chosen design solution, including your decisions, to your supervisor and/or peers and ask for feedback