COPYRIGHT

DISCLAIMER

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245

Lecture 30

Microcontroller System Design/
Mikrobeheerder Stelselontwerp

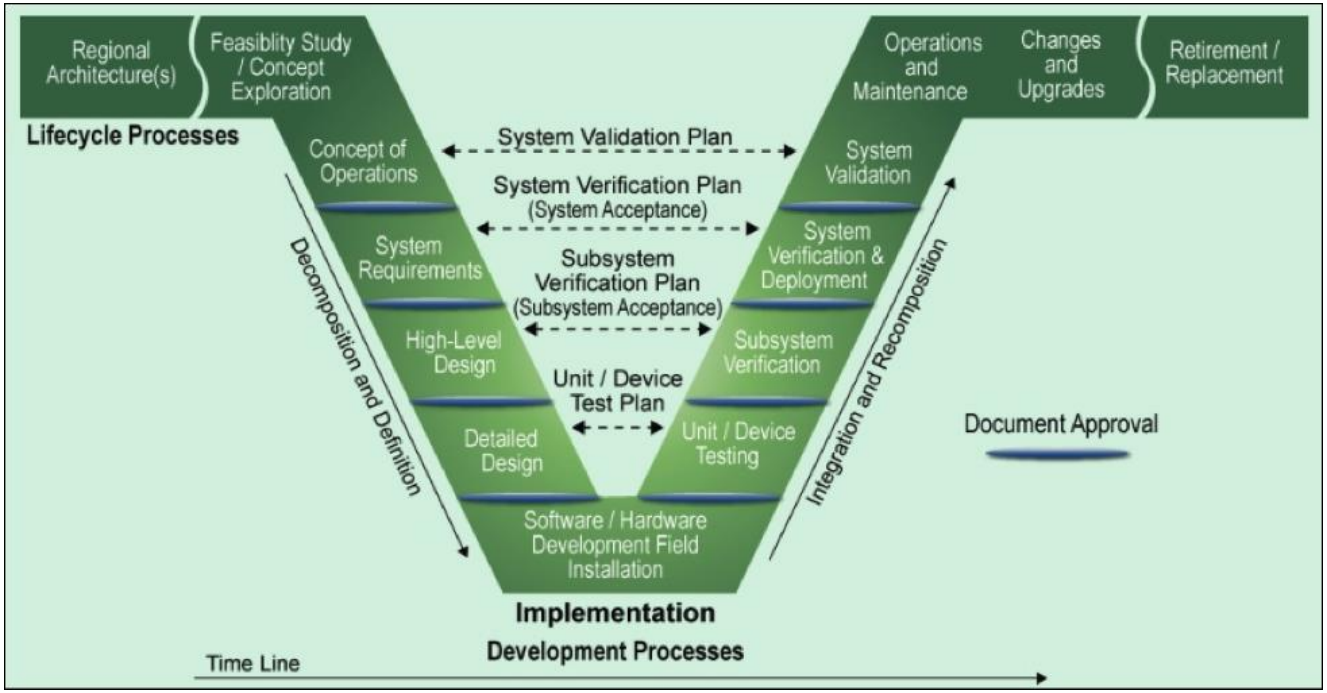
Dr Rensu Theart & Dr Lourens Visagie

Embedded Systems
Toegewyde Stelsel

- We use the term “embedded system” to describe a micro-controller based system:
- An **embedded system** is a system, consisting of a microprocessor or microcontroller and its software and accompanying electronic hardware components, that is designed to perform a **dedicated** function.
- It’s a broad term, that describes numerous different systems and applications: digital watch, medical imaging devices, satellite control system
- For our purposes, it’s the microcontroller and its peripherals, and all the supporting electronics (and the software we develop), in order to fulfil a specific purpose
- Because of the dedicated nature of such a system, the design of an embedded system is focussed around this dedicated purpose. We describe the dedicated functionality through something called “requirements”

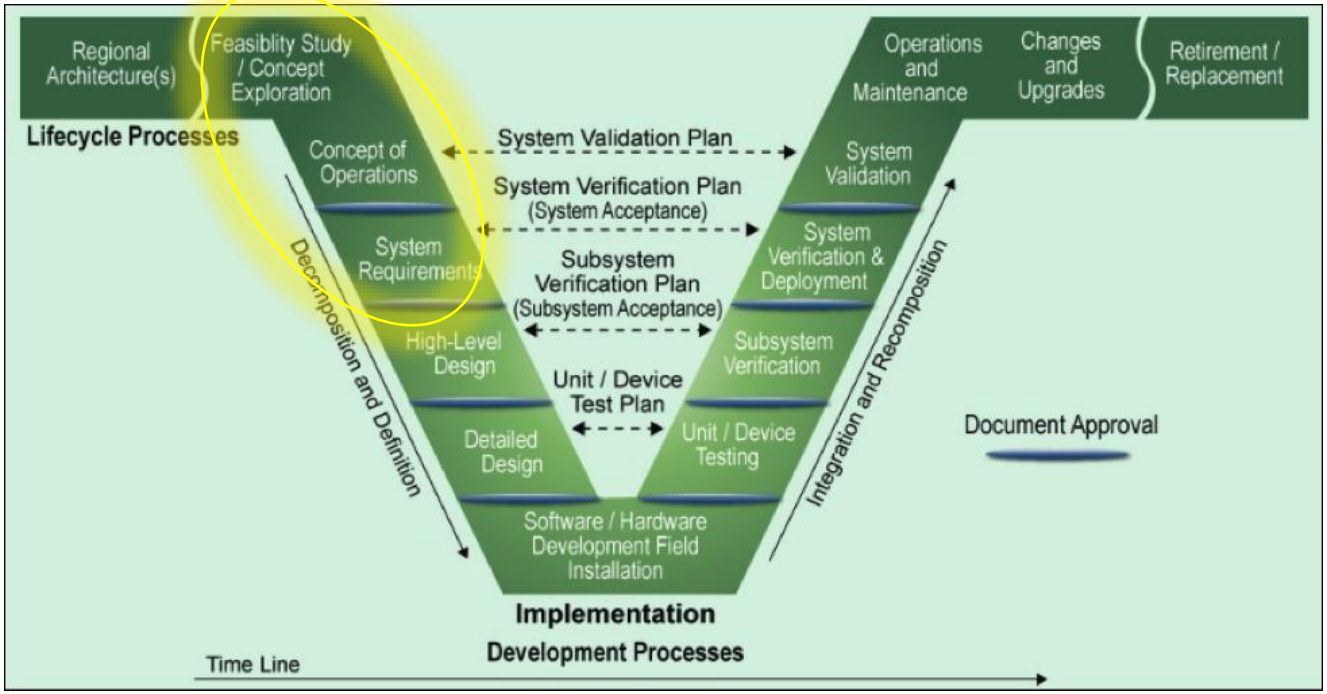
System Life Cycle
Stelsel Lewenssiklus

- System Engineering is a multi-disciplinary process that involves engineering and management



- Usually when people talk about 'Systems Engineering', it's a very formal and complex process – it generates lots of paperwork – and applied on high-cost and high profile projects.
- All embedded systems can benefit from following a systems engineering approach, but it can be implemented in a less formal, and trimmed-down way

Requirements
Vereistes



Requirements
Vereistes

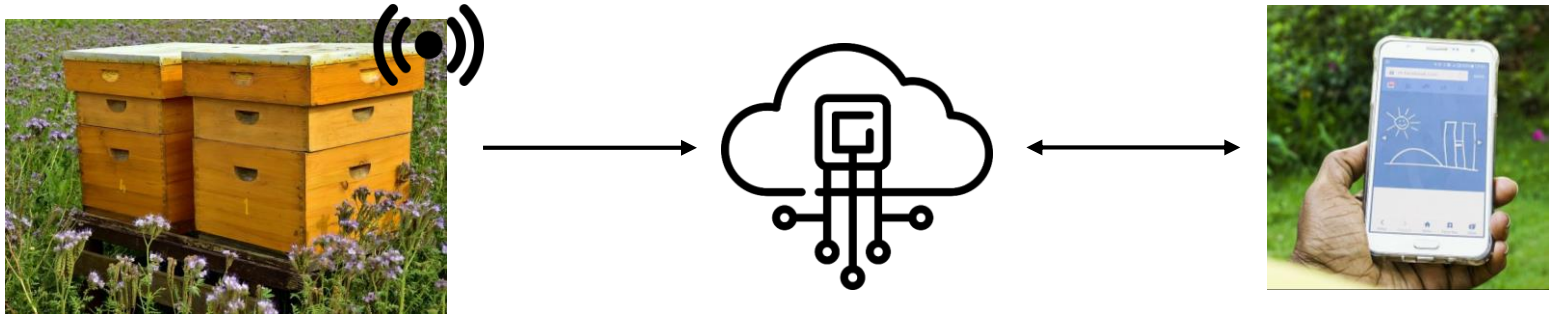
- All systems start with some 'requirements': a description of what the thing must do and/or how it must do it.
- User requirements is the description of what the user or customer wants
  - User/customer can be an external client, your boss, (yourself)
  - User/customer is not necessarily aware of the technical challenges or implications – the User Requirements are usually vague – a whish-list of what the customer wants
  - User requirements are not necessarily quantifiable or measure-able
- Then there is a process where these requirements are refined through assumption, decisions, and analysis
- This process requires your engineering knowledge
- System requirements make up the exact, unambiguous and testable description of the system.
- Usually the process of user requirements to system requirements in iterative – check with customer if this is indeed what he wants

Requirements - Example
Vereistes - voorbeeld

- Customer/your boss: “We want you to come up with a device to assist beekeepers – to prevent loss or damage to beehives”
- You: “OK… ?”
  - What exactly must it do?
  - How do you want it to work?
  - When do you want it?
  - Anything else I should know?
- Customer/boss: “Ok this is what we want:
  - It has to sense when the beehive is moved or knocked over
  - It should monitor temperature inside the beehive
  - It should not rely on external power
  - The data should be available “via the cloud”
  - The user should be notified within 1min of a movement event
  - Temperature should be recorded and logged to the cloud every hour
  - We need 1000 devices 6 months from now. It may not cost more than R300 to make one device
- The above are the **User Requirements**

Requirements - Example
Vereistes - voorbeeld

- The system components will consist of the hive sensor unit, cloud server and mobile phone app

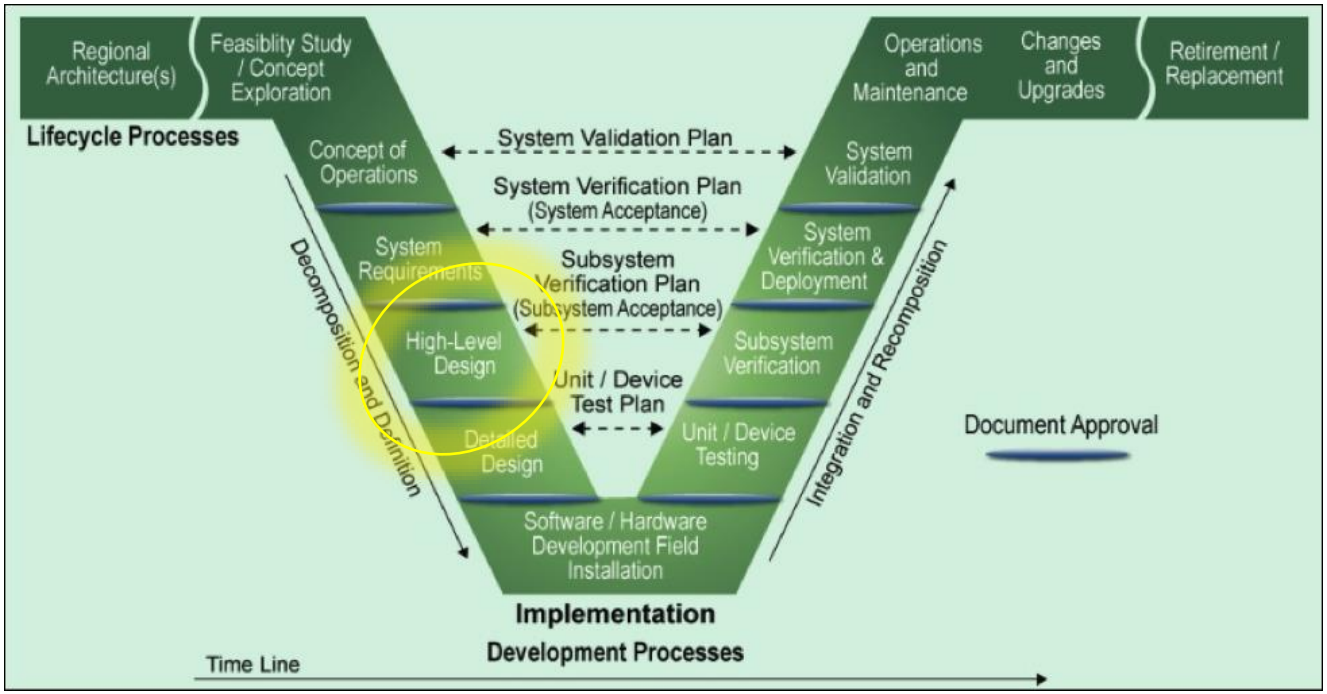


- The System Requirements for the hive sensor unit (HSU) might be something like:
  - The HSU shall register accelerations in excess of 0.1g as a movement event
  - The HSU shall measure temperature to 1°C accuracy
  - The HSU shall communicate movement events with less than 10s latency
  - The HSU shall communicate temperature measurements every 1 hour with less than 10s latency and less than 5s time drift
  - The HSU shall have an internal rechargeable battery and external solar panel
  - The HSU shall make use of an STM32F4 microcontroller (company standardizes on STM microcontrollers)
  - (other requirements for size, shape, materials etc.)

Requirements - Example
Vereistes - voorbeeld

- The use of words like “Shall”, “Must” and “Should” in requirements often convey specific meaning (but it is usually defined somewhere)
- For example:
  - MUST is equivalent to REQUIRED and SHALL indicating that the definition is an absolute requirement.
  - MUST NOT is equivalent to SHALL NOT and indicates that it is an absolute prohibition of the specs.
  - SHOULD is equivalent to RECOMMENDED means that there are valid reasons to ignore a particular requirement, but the implications need to be weighed.
  - SHOULD NOT and NOT RECOMMENDED means that a particular behavior may be acceptable or useful, but again, the implications need to be weighed.
  - MAY means OPTIONAL and that the requirement is truly optional. Interoperability with different systems that may or may not implement an optional requirement must be done.

Design
Ontwerp



Design Alternatives
Ontwerp Alternatiewe

- The design engineer has to select between different ways of achieving the desired functionality (satisfying the requirements)
- The choice usually involves comparison and "trade-off" – prioritising one aspect over another. Properties that are compared include cost, complexity, availability, development time

Design Alternatives
Ontwerp Alternatiewe

- There are different IoT technologies that we can use for the HSU data link:
  - Cellphone data connection:
    - Makes use of the normal cellphone network. As long your cellphone shows coverage, the HSU will be able to connect.
    - HSU will need GPRS modem, a SIM card, and monthly subscription charges apply
  - SigFox:
    - Proprietary global network. Fairly good urban coverage (but maybe not in remote areas?). Makes use of low-power long-range transmissions with low data rates.
    - HSU will need a SigFox modem, and monthly subscription charges apply
  - LoRa
    - Also a low-power wide-area network intended for low data rate transmission. There is something called “The Things Network” which is an open network – coverage depends on people putting up “gateways”. In remote areas, more likely you will have to put up your own gateway (cost involved) and the gateway will have a higher bandwidth connection to the internet.
    - HSU will need a LoRa modem. Per device no subscription applies, but the gateway will need an internet connection with monthly subscription charges

Design Alternatives
Ontwerp Alternatiewe

Alternative	Cost			Coverage[1]	Development time
	Components (Per HSU)	Monthly subs.	Gateway (once-off)		
Cellphone GPRS	R200	R15	-	Good	High
SigFox	R250	R10 [3]	-	Good	Low
LoRa	R250	R5 [4]	R10000	None[2]	Medium

1. In target deployment area
2. Will require a LoRa gatewat to be setup
3. Cost per unit, assuming 1000 units
4. Gateway internet connection, split across 1000 units

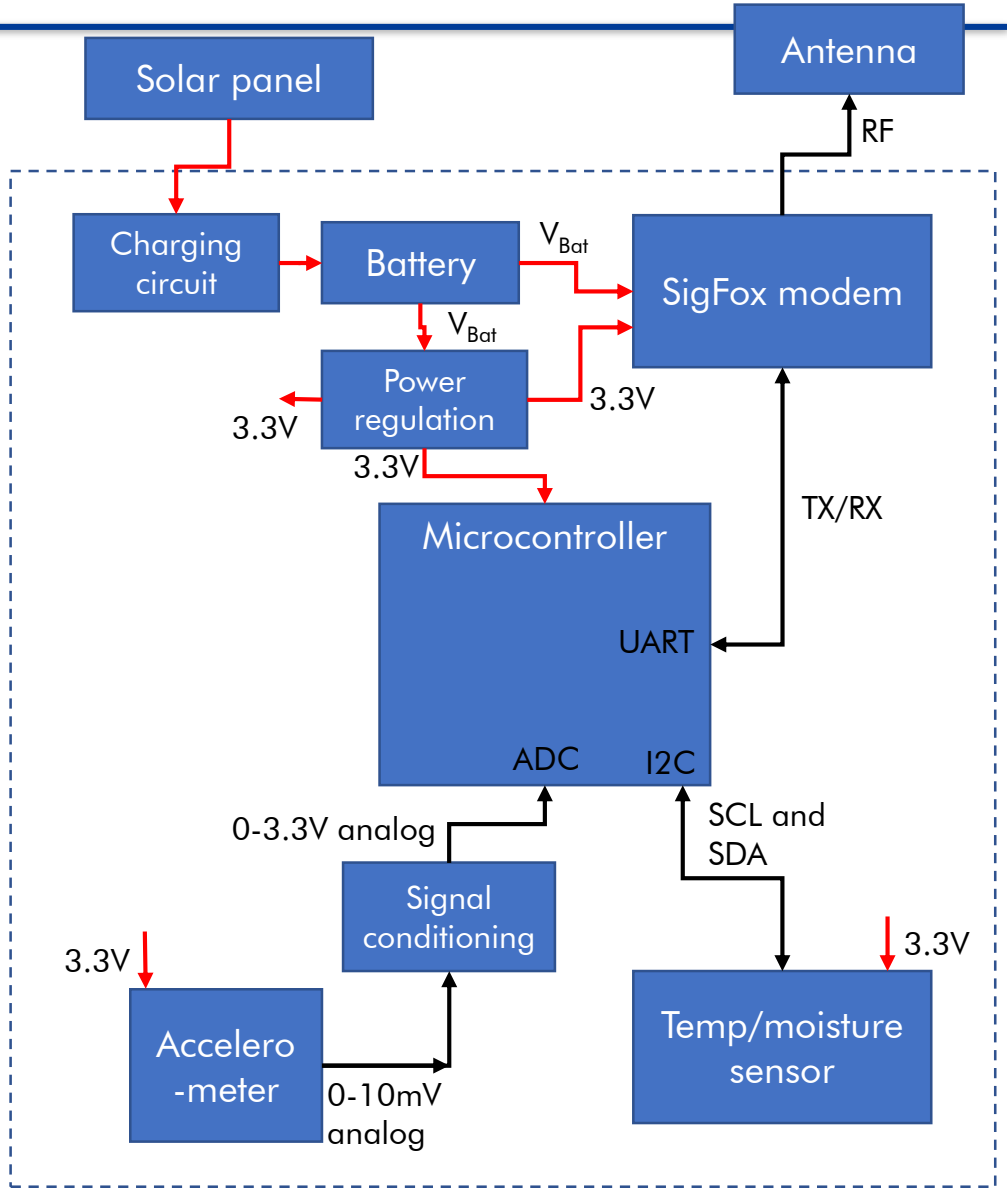
Concept Design
Konsep Ontwerp

- Identify components and their dependencies
- Block diagrams help to visualise the design
  - Each component = block in the diagram
  - Blocks connect via interfaces – lines in the diagram
    - Label the signal types, and peripherals they connect to
    - Use arrows to indicate flow of data (not necessarily current)

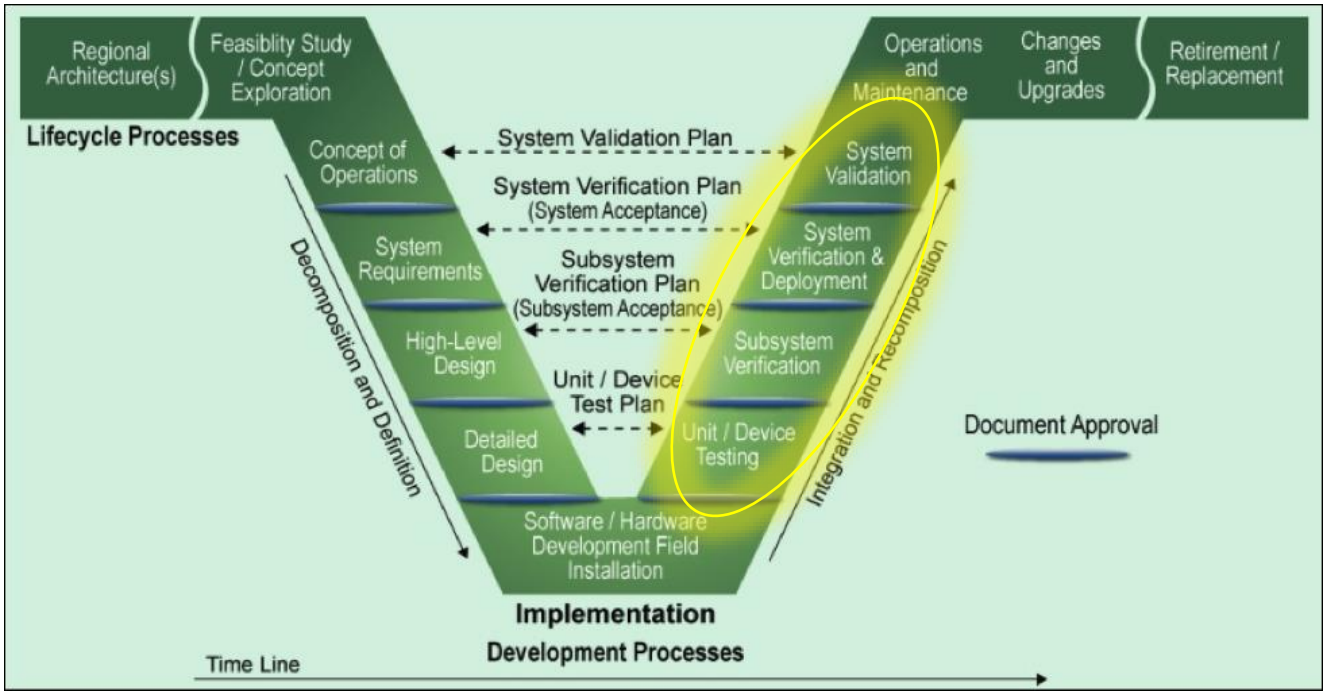
Concept Design
Konsep Ontwerp

Example:

- The HSU shall make use of a STM32F411VE microcontroller
- The HSU shall measure acceleration using a … accelerometer with an analog output: 0 to 1g = 0 to 10mV
- The HSU shall measure temperature and moisture using a BME280 sensor – I2C interface
- The HSU shall communicate using a SigFOx modem – UART interface to microcontroller
- The SigFox modem requires an external passive antenna (passive = antenna does not need power)
- The HSU shall make use of internal timer peripheral to measure time
- The HSU shall per powered from a rechargeable 7.4V LiPo battery
- The HSU battery shall be charged by external solar panel
- The MCU, accelerometer, temperature/moisture sensor requires 3.3V regulated power
- The SigFox modem requires both 3.3V regulated and 7.4V unregulated (battery) power

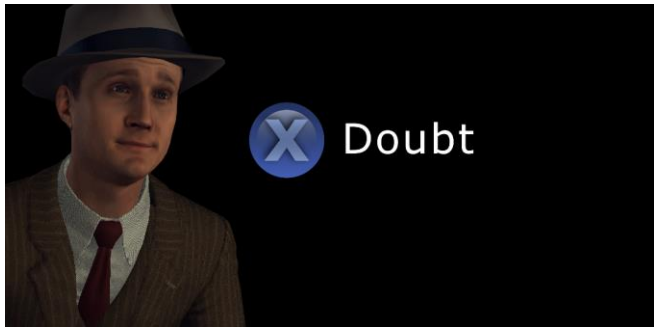


Verification
Verifikasie



Verification
Verifikasie

- "Ek het hom getoets, hy werk"…
- "I've tested it. It works"…



- How did you test?
- Why can you say that "it works"?

- The reason why requirements are exact, unambiguous, specific, is that we have to devise our tests to make sure the requirements are met
- The process is called "verification": Make sure that the component, or system complies with a requirement, specification, or imposed condition (make sure that whatever we've implemented does what it is supposed to)
- Sometimes it involves testing, but could also be other methods
  - Inspection
  - Demonstration
  - Test
  - Analysis
- "Requirements verification matrix" is a table that documents how you are going to verify each requirement
- The verification strategy is something that you decide on even before implementing

Testing and Measurement
Toetse en metings

- Tests should be able to verify the specifics of the requirement – 1degC, 0.1g acceleration. 10s latency
- Tests should have
  - List/mention the exact equipment that is needed (multimeter, oscilloscope, logic analyser)
  - Detailed procedure – so that someone else can reproduce/perform the test
  - Test results themselves
  - Interpretation of the results, in order to justify why the requirements are met

Testing and Measurement Example
Toetse en metings Voorbeeld

- The power supply has to supply power to the following components:

Component	Supply voltage	Current used
Microcontroller	3.3V	100mA
SigFox modem	7.4V	50mA when transmitting, 1mA when idle
Accelerometer	3.3V	5mA
Temp/humidity sensor	3.3V	5mA

- Therefore max 3.3V load current is 110mA.

BAD:

- Method: Measure the 3.3V voltage
- Results: 3.3V measurement is 3.2V
- Conclusion: I've tested it, it works!

Good:

- Method:
  - Switch off power to the power supply
  - Connect a 30 ohm load resistor with 500mW power rating to terminal J12, pins 1 and 2
  - Connect oscilloscope probes to J13, pins 1 and 2
  - Switch on power switch S1
  - Record the minimum, maximum and average voltage measured by the oscilloscope
- Results: The tabulated measurements can be seen in table X, and fig Y
- Conclusion: The power supply is capable of supplying 3.3V power at maximum load, with a maximum deviation of 0.1V peak-to-peak.