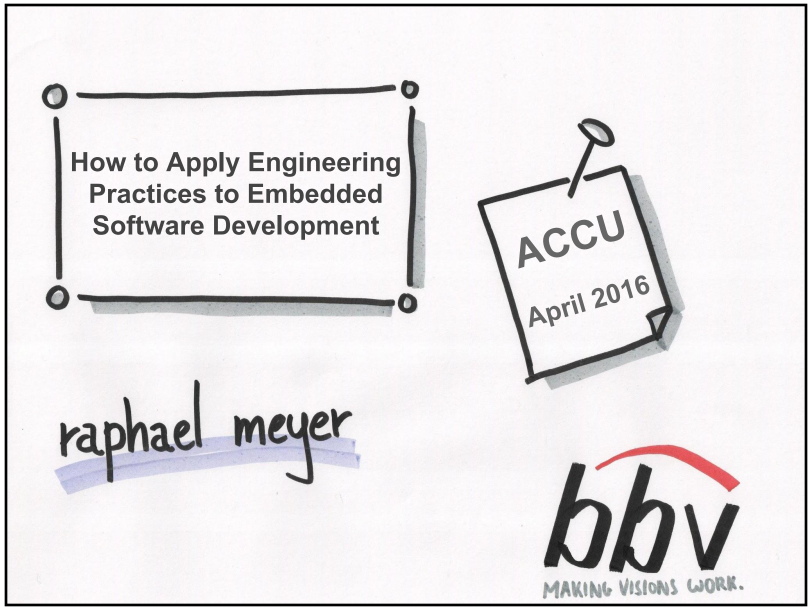


How to Apply Engineering Practices to Embedded Software Development

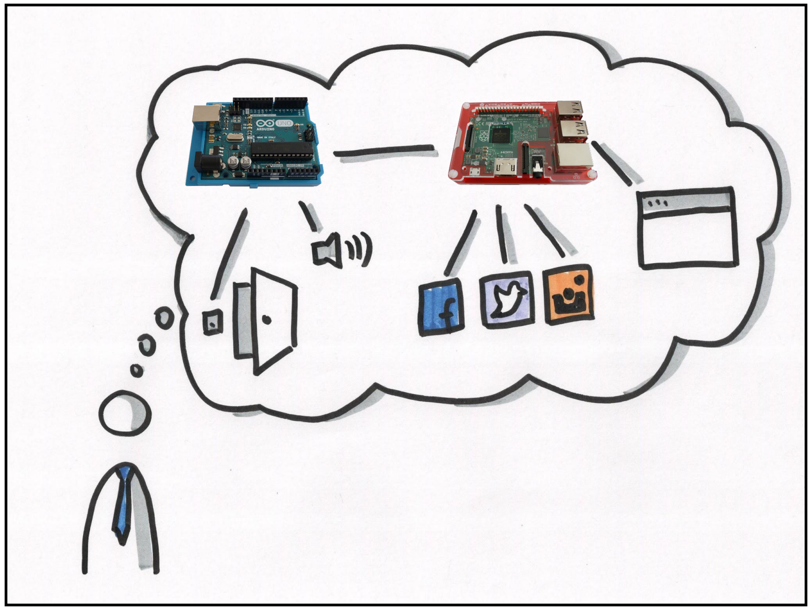
ACCU

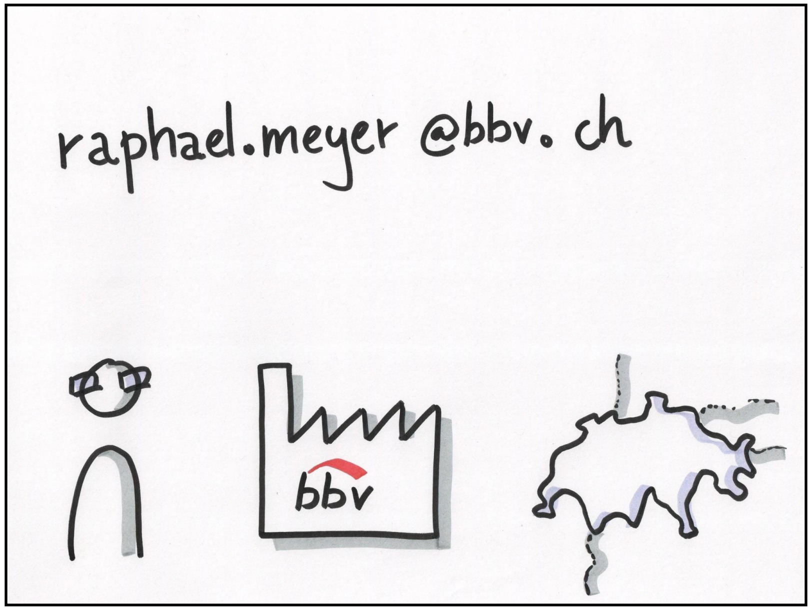
April 2016

raphael meyer

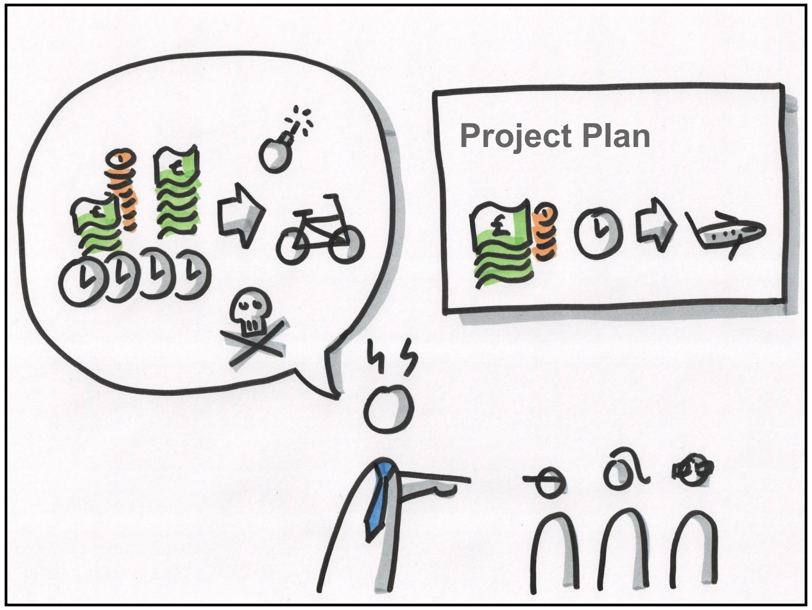
bbv

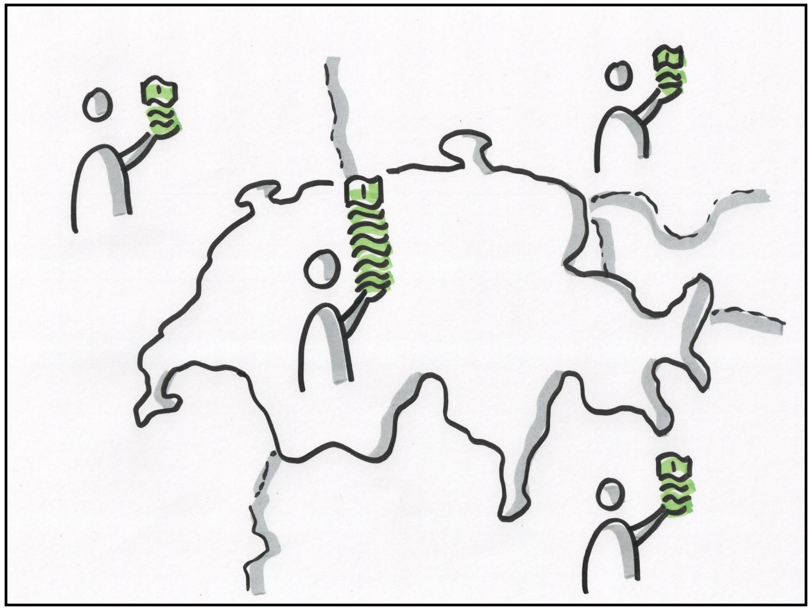
MAKING VISIONS WORK.

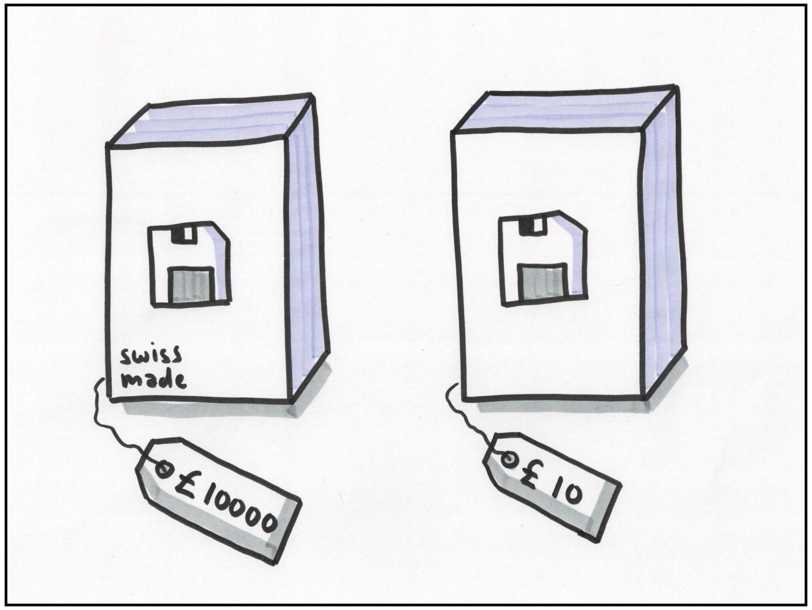


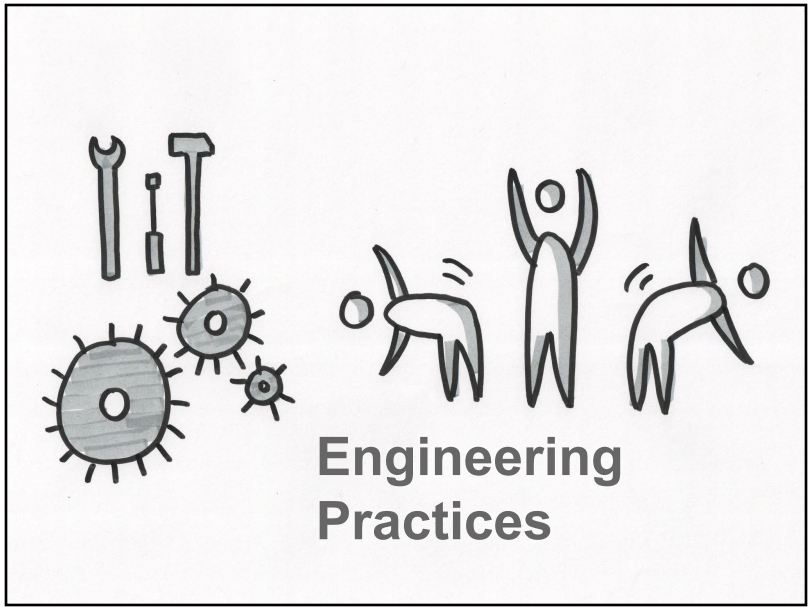


raphael.meyer@bbv.ch

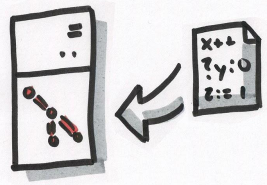




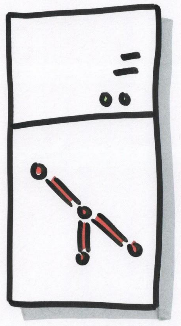




Version Control System



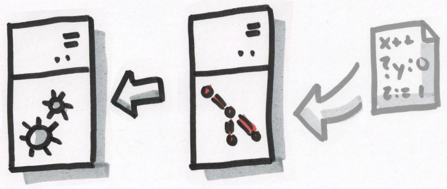
Version Control System



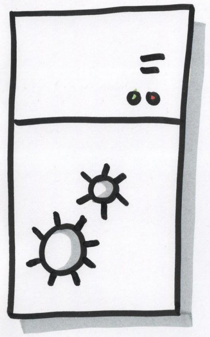
GitHub

git

Continuous Integration

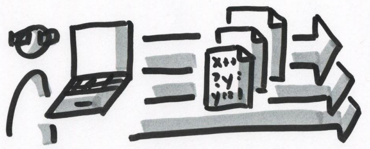


Continuous Integration



Jenkins

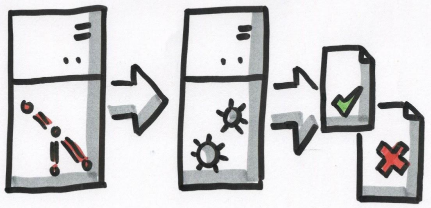
Travis CI



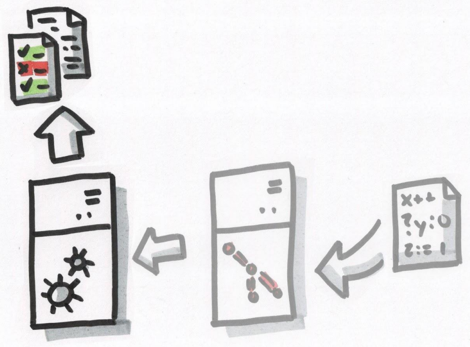
Continuous Integration is not about a tool.

It's the practice of frequently publishing your changes to a shared repository.

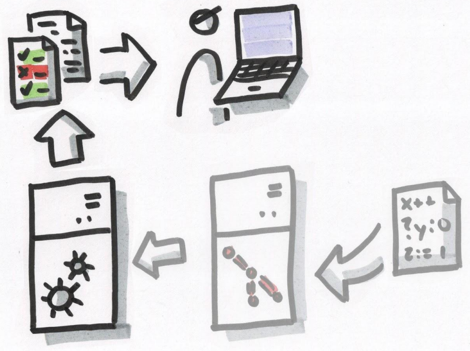
The tools that are associated with Continuous Integration do the verification of each integration to detect errors as quickly as possible.



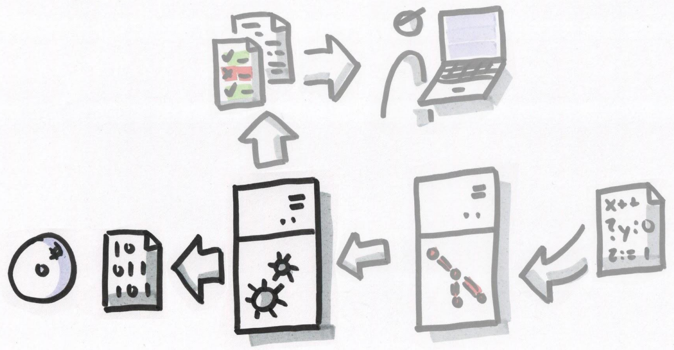
Continuous Integration



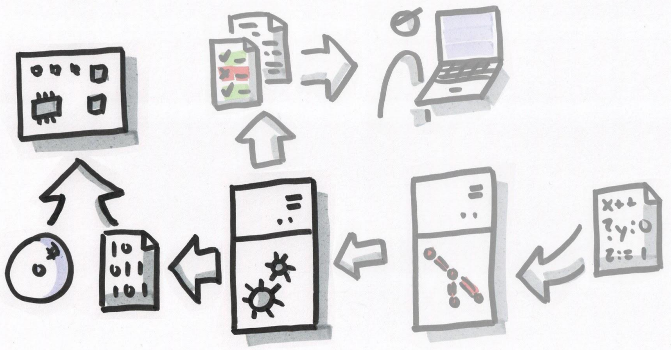
Continuous Integration



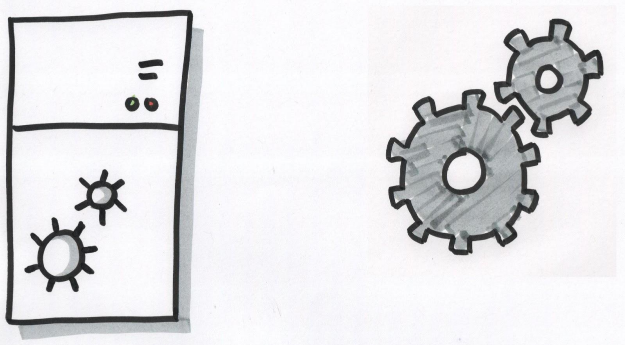
Continuous Integration



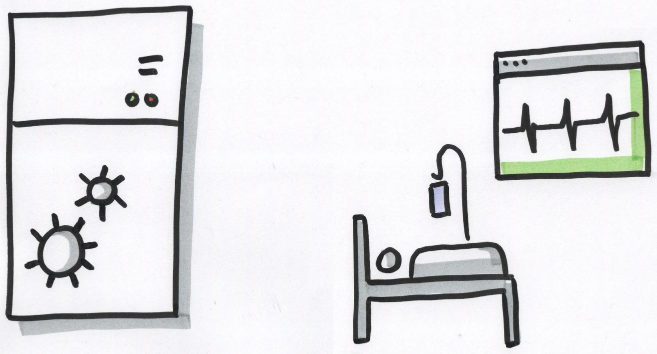
Continuous Integration



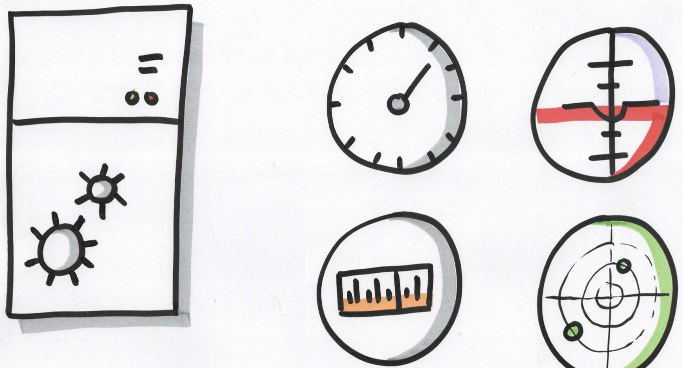
Continuous Integration



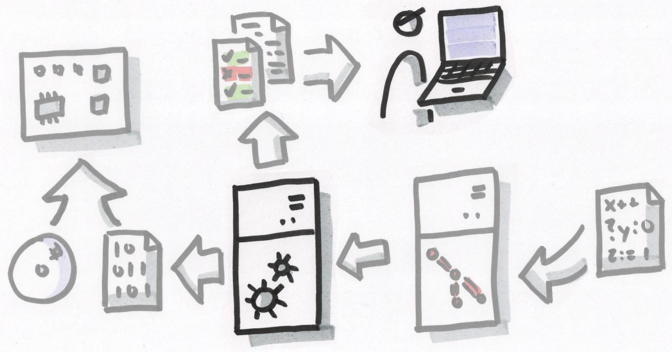
Continuous Integration

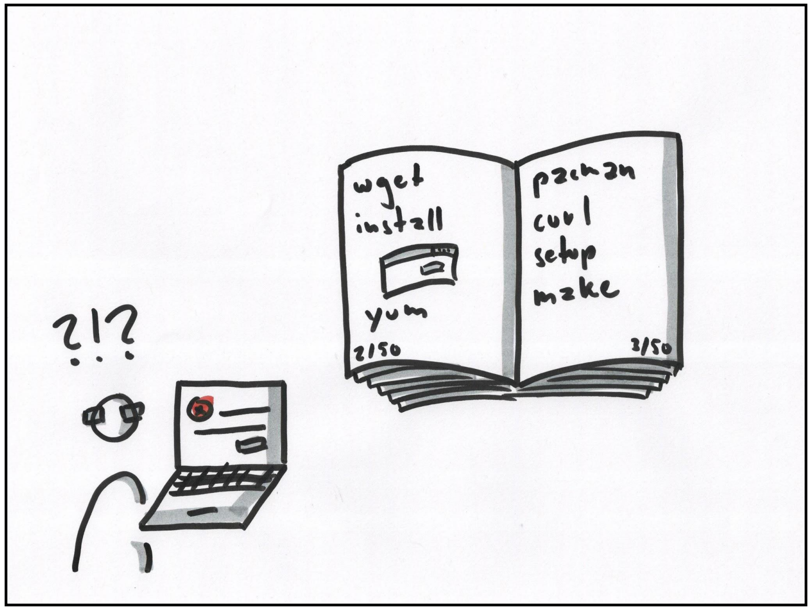


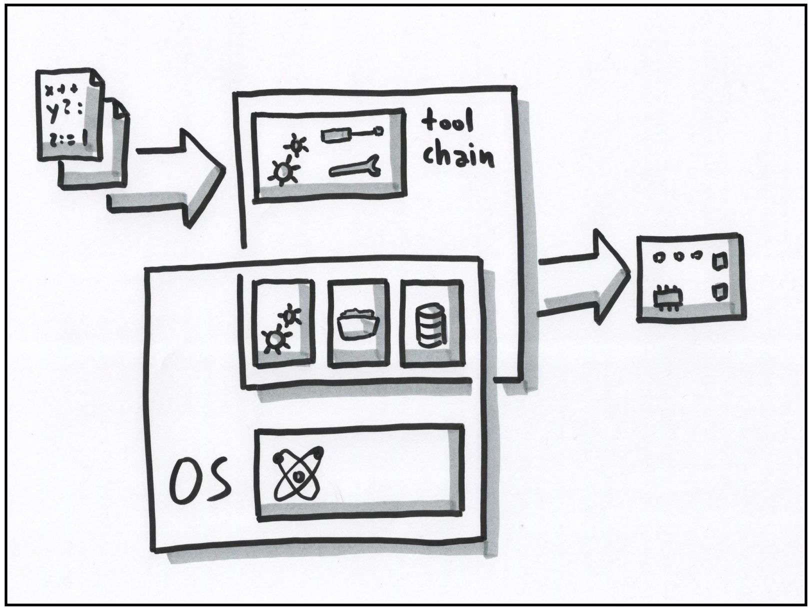
Continuous Integration

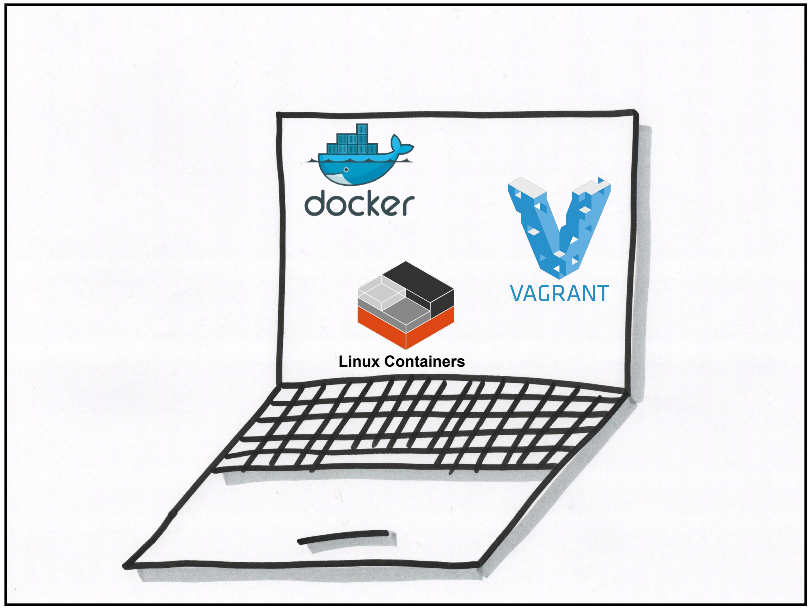


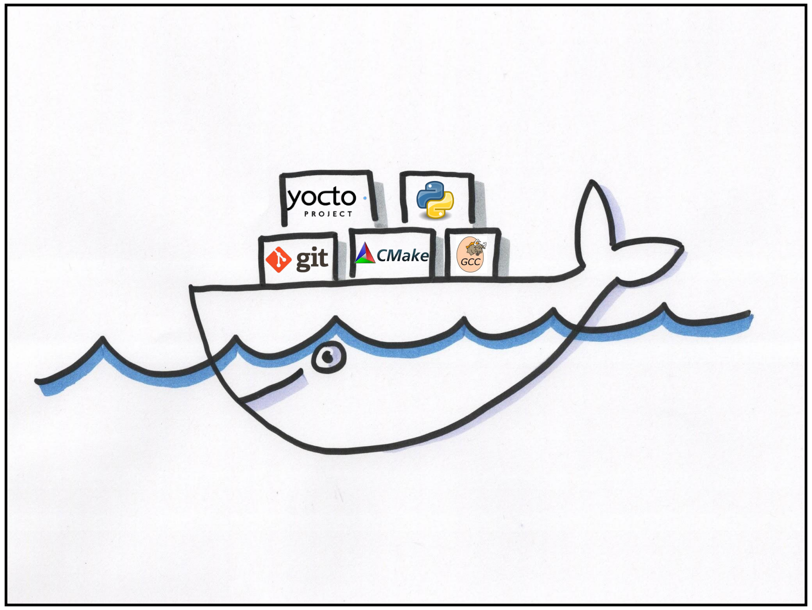
Toolchain

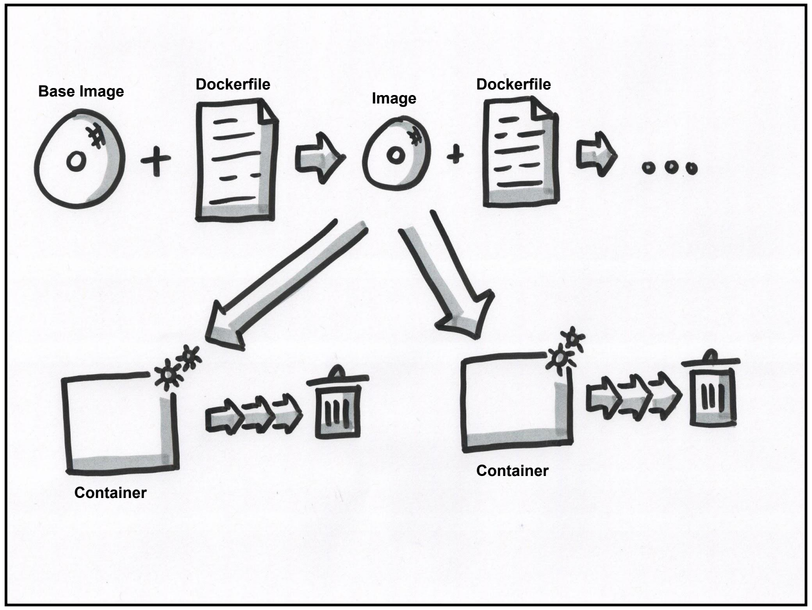


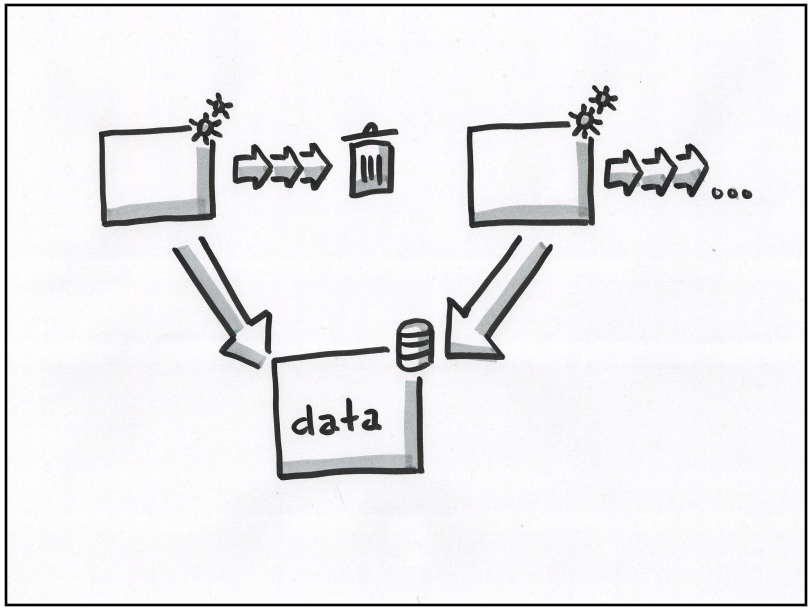

?!?
wget
install
yum
2/50
pacman
curl
setup
make
3/50

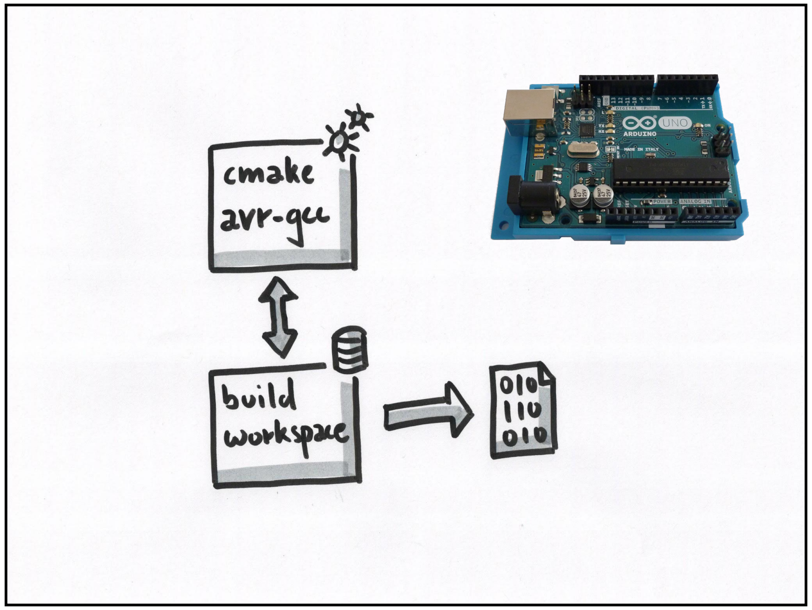

tool chain
OS

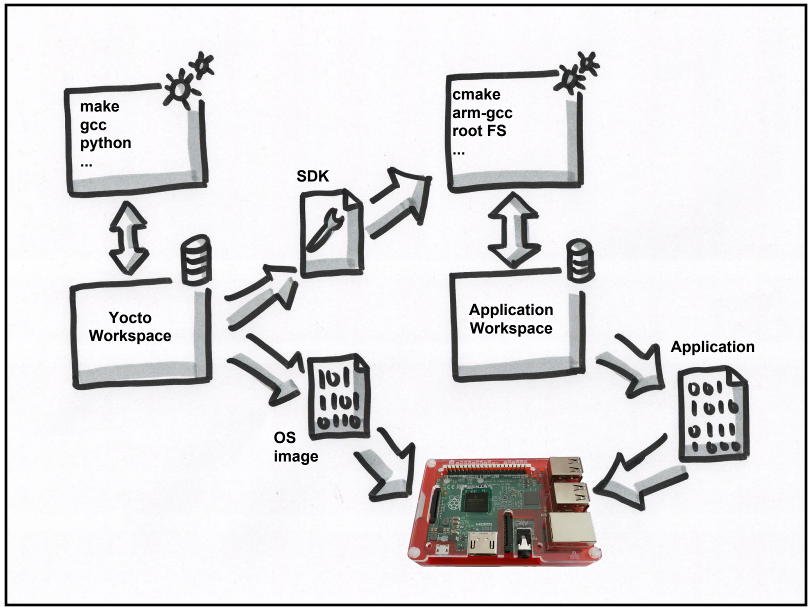

docker
VAGRANT
Linux Containers


yocto PROJECT
git
CMake
GCC

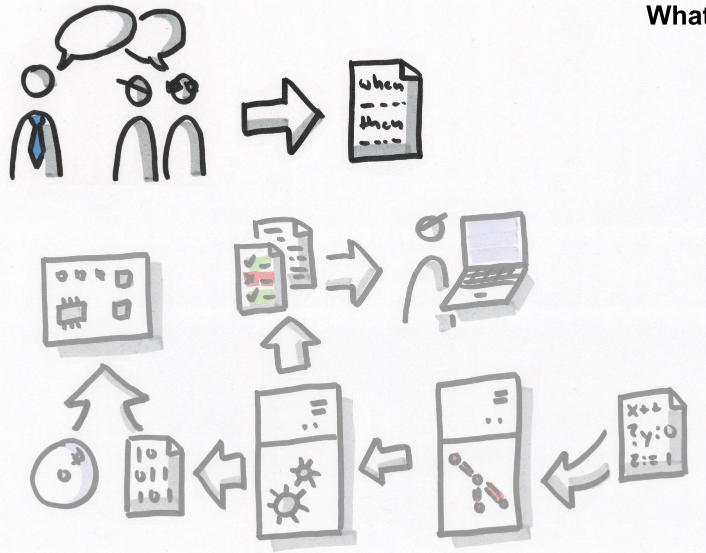

Base Image
Dockerfile
Image
Dockerfile
Container
Container

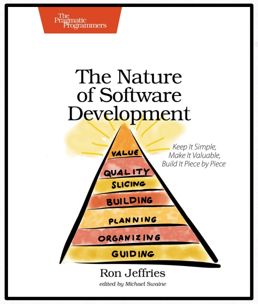

data


cmake
avr-gcc
build
workspace
010
110
010


make
gcc
python
...
cmake
arm-gcc
root FS
...
SDK
Yocto
Workspace
Application
Workspace
Application
OS
image

What?




The Pragmatic Programmers

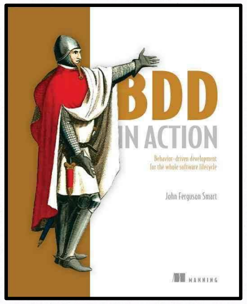
The Nature of Software Development

Keep It Simple, Make It Valuable, Build It Piece by Piece

VALUE
QUALITY
SLICING
BUILDING
PLANNING
ORGANIZING
GUIDING

Ron Jeffries

edited by Michael Swaine

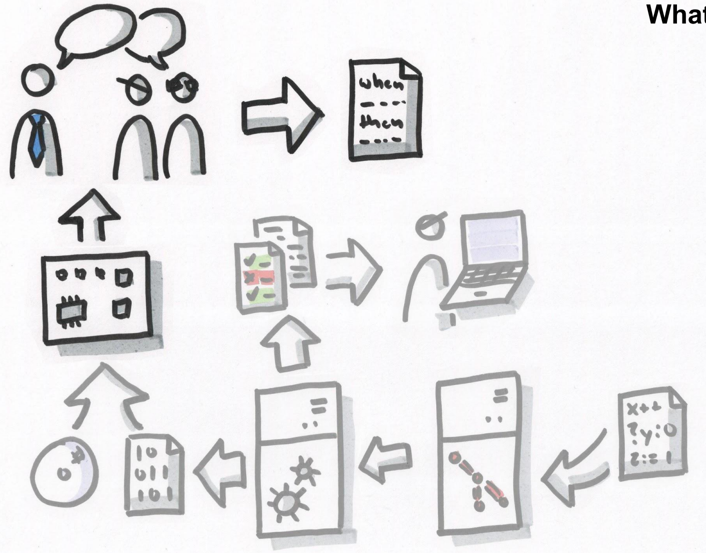

BDD IN ACTION

Behavior-driven development for the whole software lifecycle

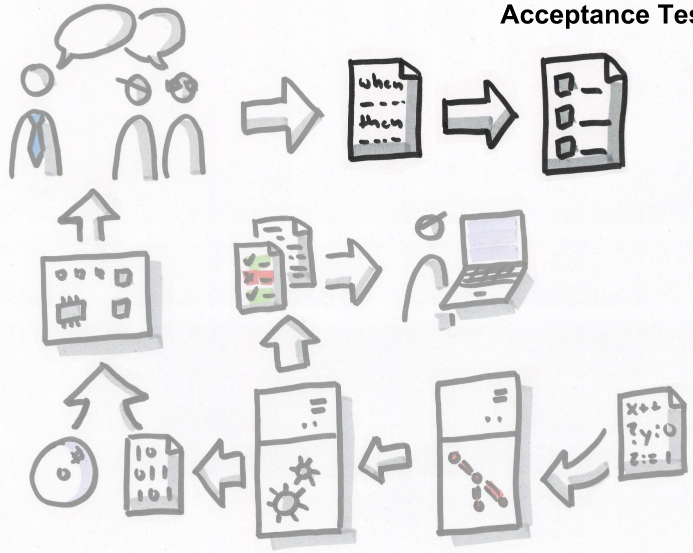
John Ferguson Smart

MANNING

What?



Acceptance Tests



Acceptance Tests

… define when a feature is done.

… ensure that stakeholders, testers and developers all understand what the desired system behaviour is.

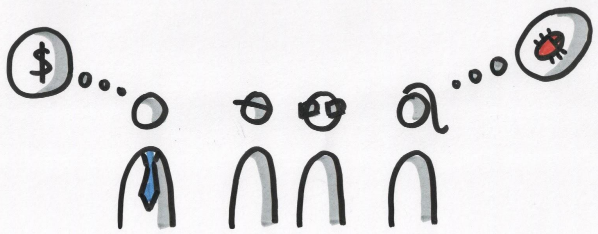
… are automated as part of the continuous integration system.

Acceptance Tests

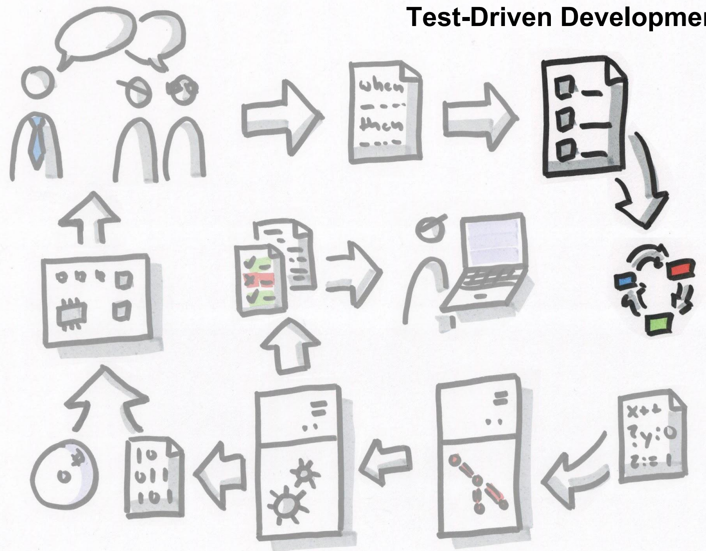
«Typically **business analysts** write the "**happy path**" versions of the tests, because those tests describe the features that have business value.

QA typically writes the "**unhappy path**" tests, the **boundary conditions**, **exceptions**, and **corner cases**. This is because QA's job is to help think about what can go wrong.»

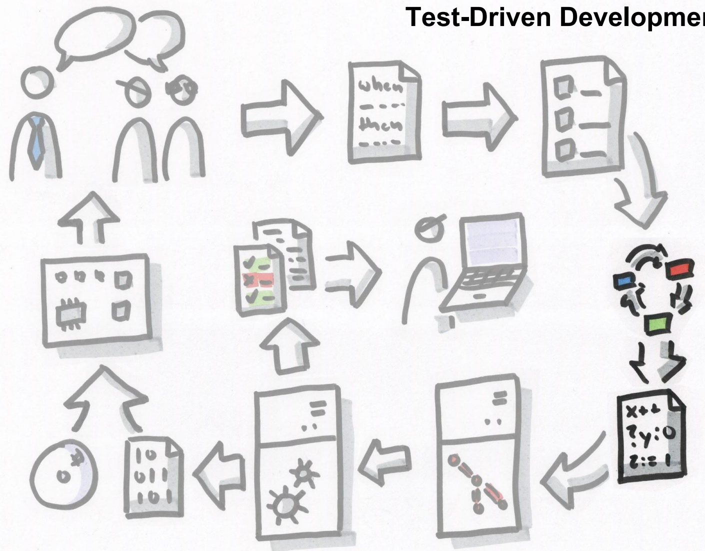
Robert Martin, The Clean Coder

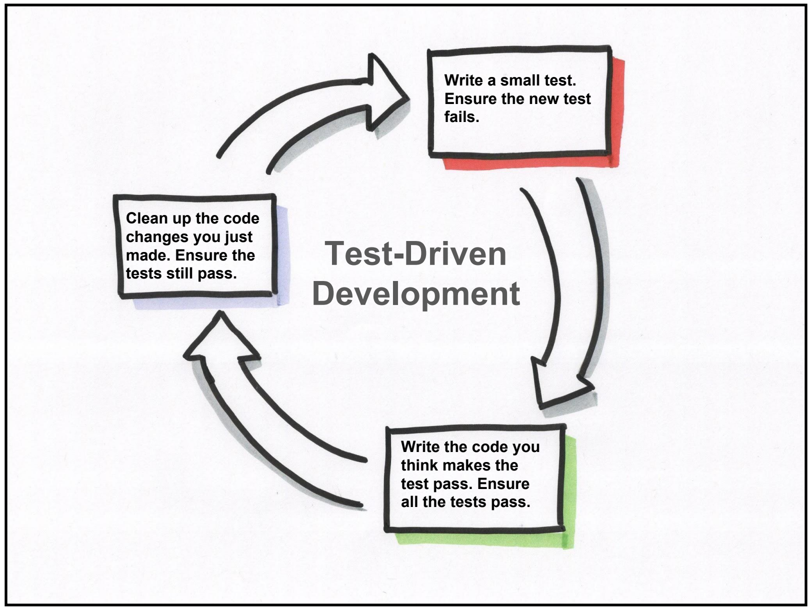


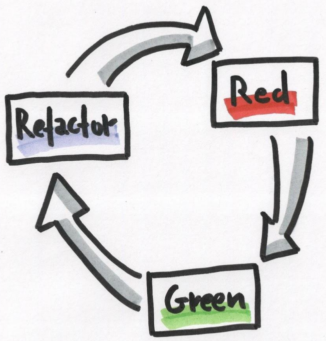
Test-Driven Development

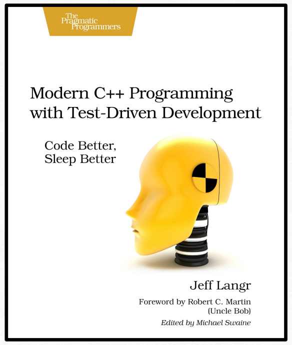


Test-Driven Development




Write a small test. Ensure the new test fails.
Write the code you think makes the test pass. Ensure all the tests pass.
Clean up the code changes you just made. Ensure the tests still pass.
Test-Driven Development




The Pragmatic Programmers

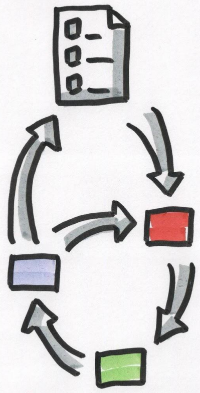
Modern C++ Programming with Test-Driven Development

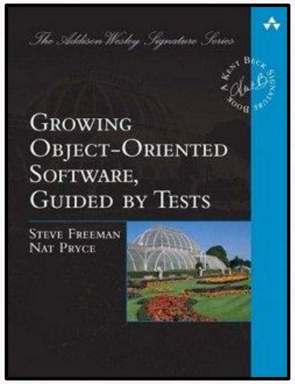
Code Better, Sleep Better

Jeff Langr

Foreword by Robert C. Martin (Uncle Bob)

Edited by Michael Swaine



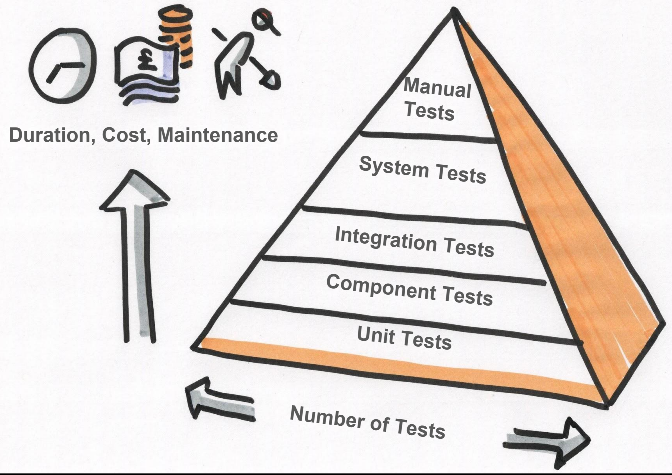

The Addison-Wesley Signature Series

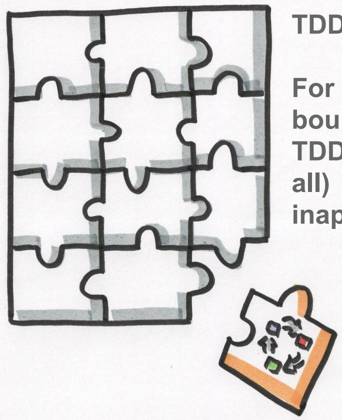
A Kent Beck Signature Book

Growing Object-Oriented Software, Guided by Tests

Steve Freeman
Nat Pryce

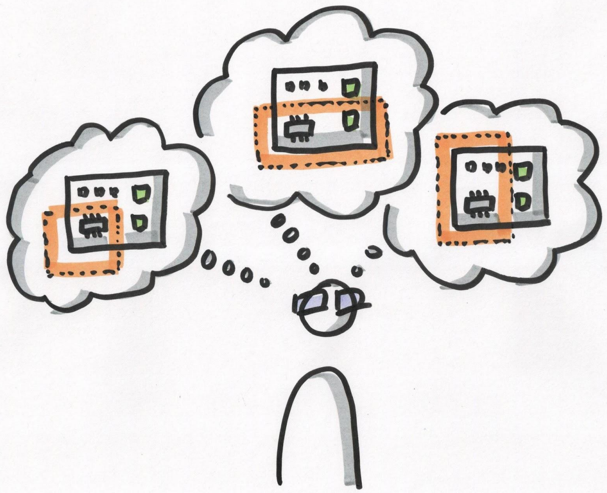
Test Pyramid

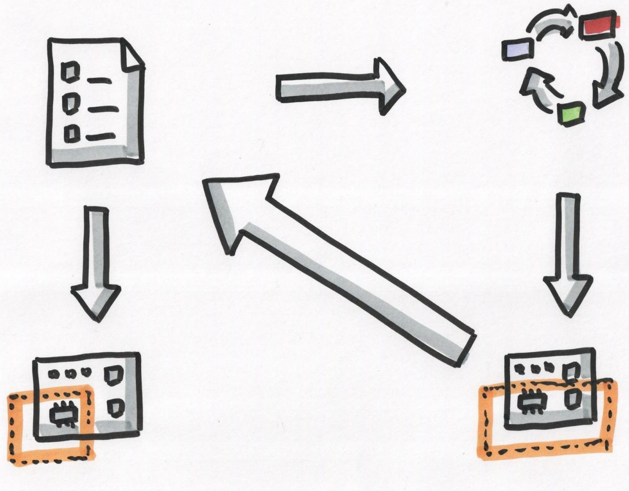




TDD does not always work.

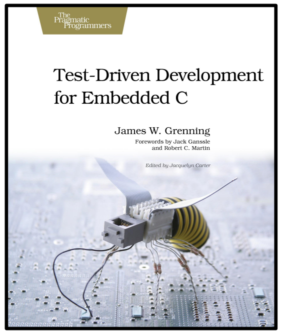
For example at the physical boundaries of the system, TDD (or automated tests at all) may be impractical or inappropriate.

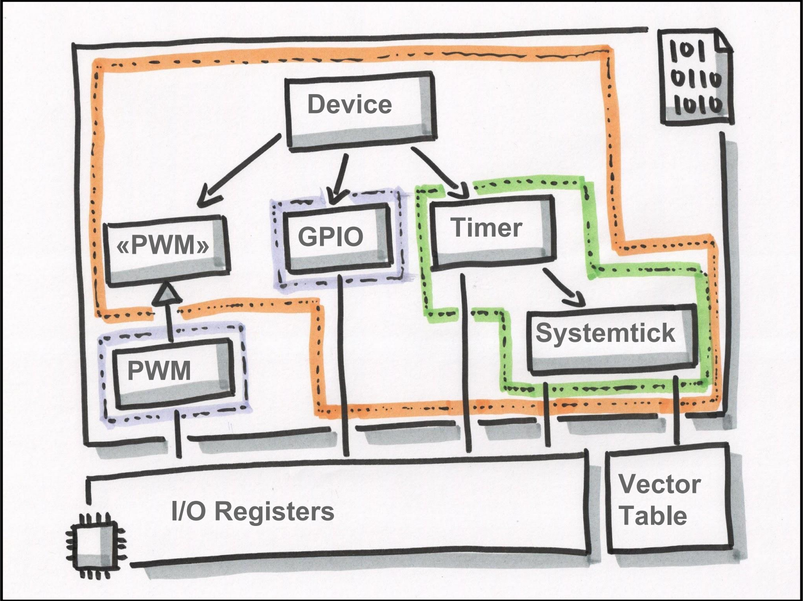


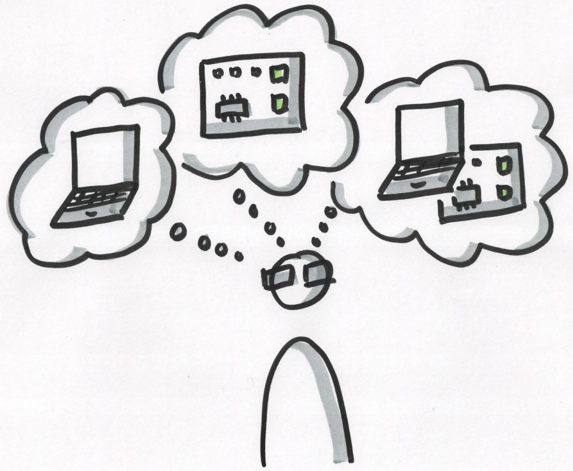


Test Doubles

```
TEST(An_output_gpio, is_low_after_configuration)
{
  uint8_t volatile & port = PORTD;
  uint8_t const pin = 3;

  // Arrange
  Gpio testee;
  Gpio_init(&testee, Port_D, Pin_3);
  port = 0xFF;

  // Act
  Gpio_set_direction(&testee, Direction_Output);

  // Assert
  ASSERT_THAT(port & (1 << pin), Eq(0));
}
```


The Pragmatic Programmers
Test-Driven Development for Embedded C
James W. Grenning
Forewords by Jack Ganssle and Robert C. Martin
Edited by Jacquelyn Carter





Run tests

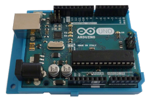
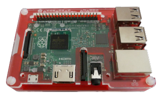
… on **host**,
and use **tests doubles** for hardware interaction.

- Use for test-driving

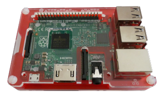
Run tests

…**eval board**, **target hardware**,
and use **tests doubles** for hardware interaction.

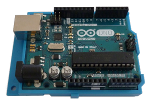
- Compiler compatibility check

Run tests



… on **eval board**, **target hardware**,
and and interact with **actual hardware**.

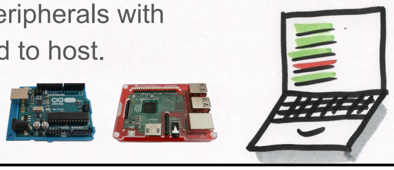


- Basic driver testing
- Playground for exploring hardware
- Suitable for peripherals with no external input, e.g. RTC, EEPROM
- Extendable by adding loopbacks, e.g. GPIO
- Extendable by adding hardware for inputs, e.g. ADC

Run tests

… on **host**, interacting with “public” interfaces of **eval board**, **target hardware**.

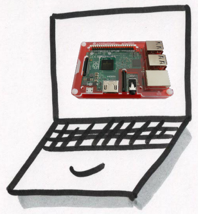
- Acceptance tests, system tests, integration tests etc.
- Tests on host may use different language.
- Requires the product to have accessible interfaces, e.g. network connectivity …
- … or custom hardware to interact with peripherals.
- Possibly replace peripherals with adapters connected to host.



Run tests

… in **emulator**,
and use **tests doubles** for hardware interaction.

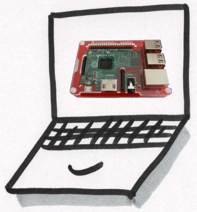
- Sometimes emulator already available, e.g. QEMU in Yocto.

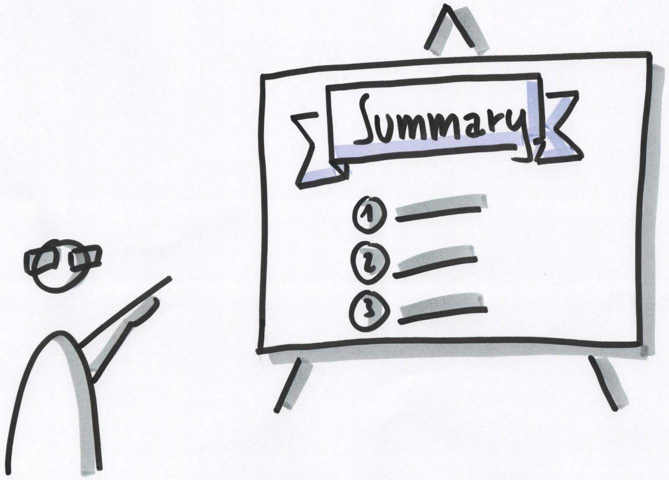


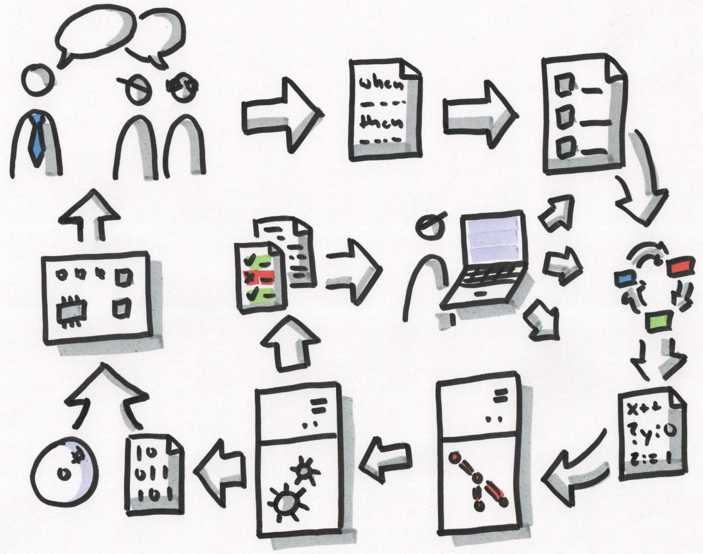
Run tests

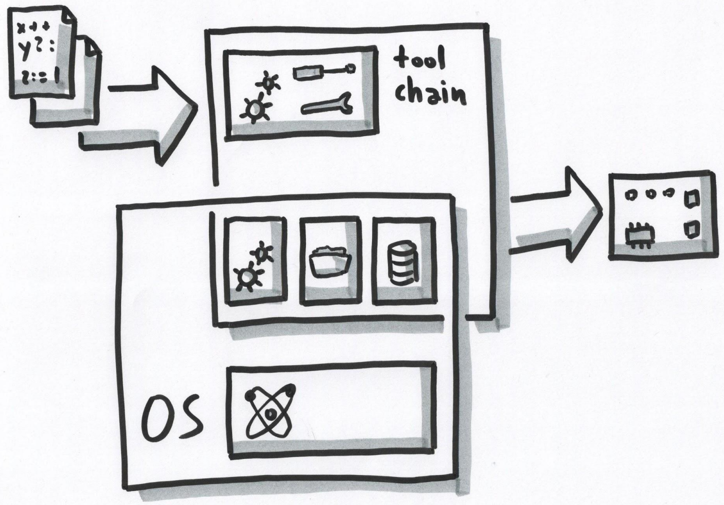
… in **emulator**,
and interact with **emulated hardware**.

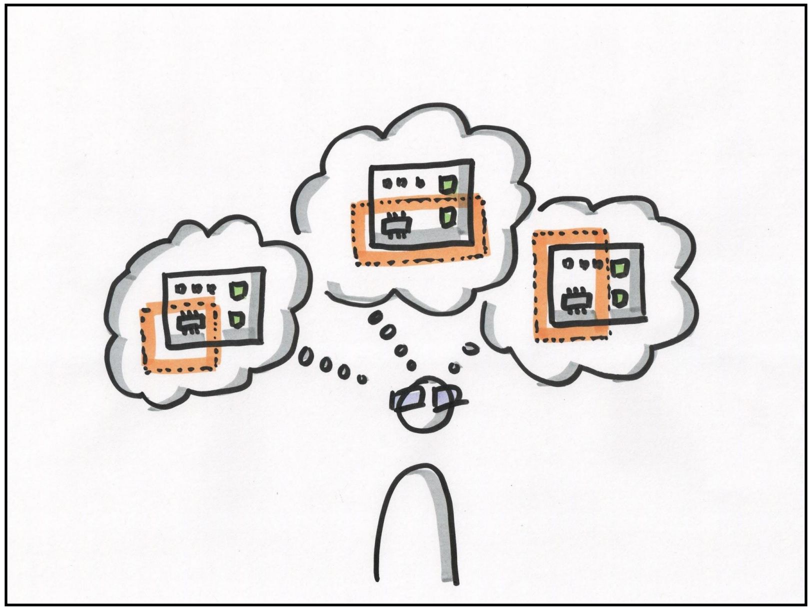
- Helpful under certain conditions
- High costs to set up
- High maintenance costs if target hardware often changes

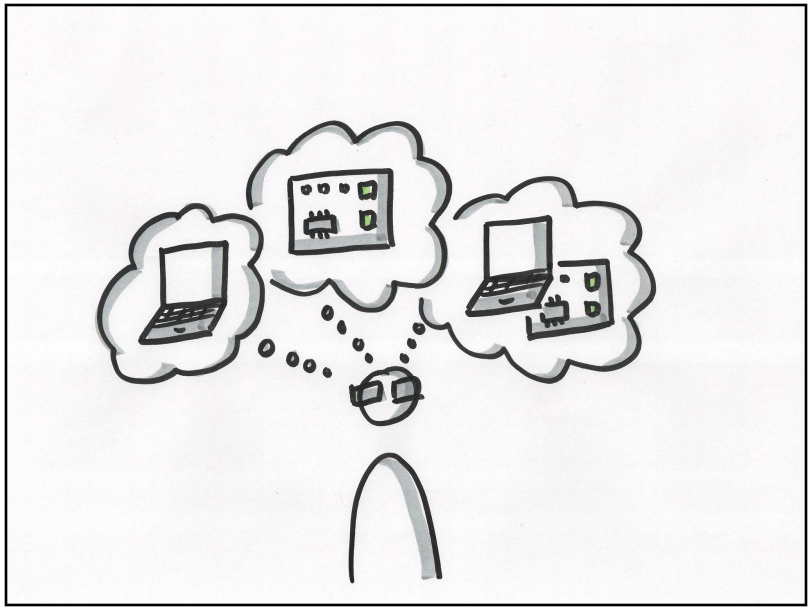




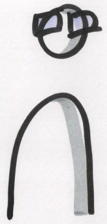








raphael.meyer@bbv.ch



bbv

MAKING VISIONS WORK.

Growing Object-Oriented Software, Guided by Tests
Steve Freeman, Nat Pryce

Test-Driven Development for Embedded C
James W. Grenning

Modern C++ Programming with Test-Driven Development
Jeff Langr

The Clean Coder
Robert C. Martin

The Nature of Software Development
Ron Jeffries

BDD in Action
John Ferguson Smart

The Pragmatic Programmer
Andrew Hunt, David Thomas