mind
embedded development
an ESSENSIUM division

Testing embedded software

ESSENSIUM

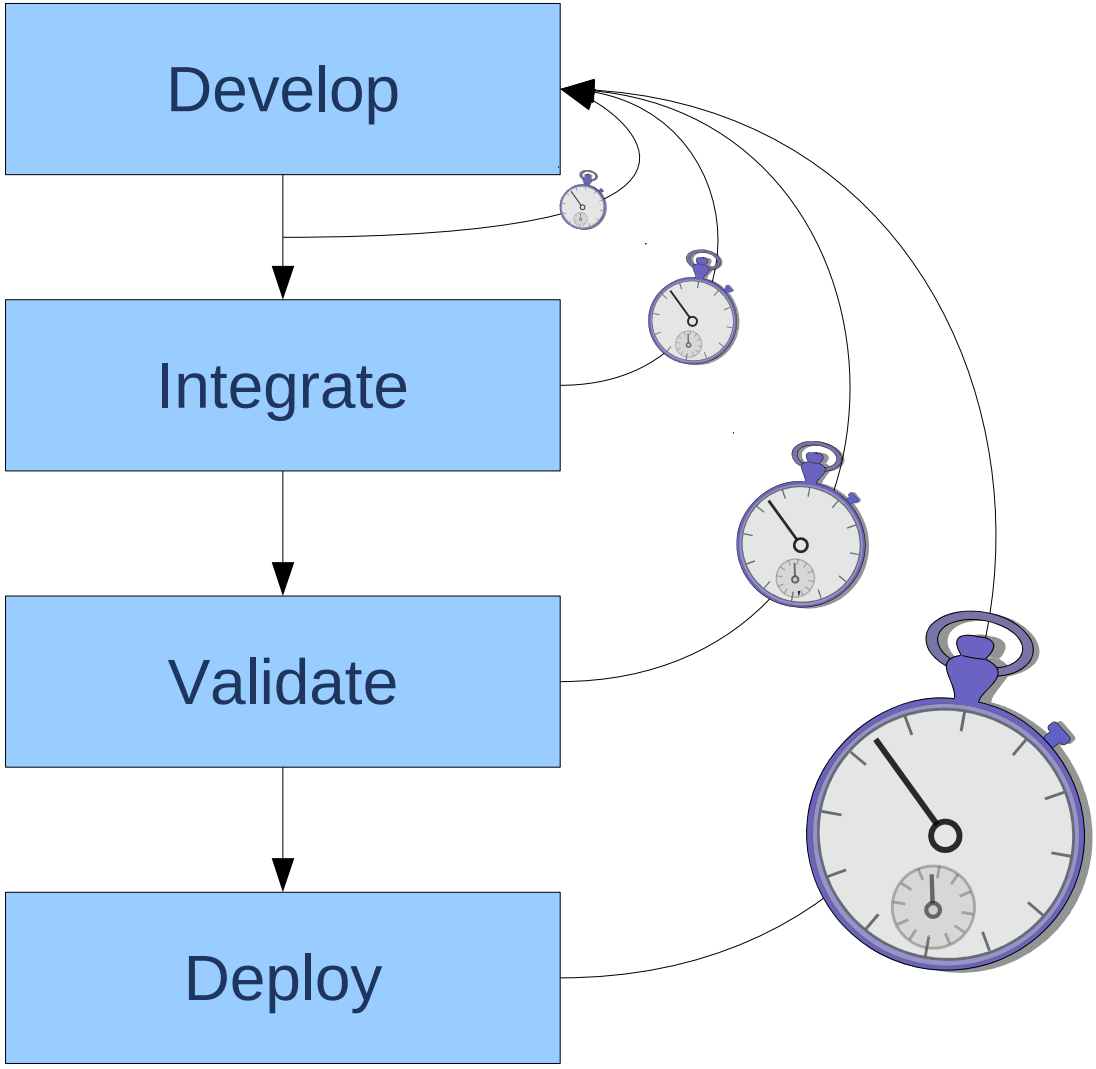
Overview

1. Testing =
   Efficient software development

2. Testing embedded software =
   special

3. Open source =
   more testing?

Testing is omnipresent in the software development process



Writing tests saves time

- Development is modifying code
- Every change risks breaking the code
- The longer it takes to find a problem, the more costly to fix
- Write tests early, to save time later
  - More difficult to add test to existing software

Think of the future

- You will modify your code
- Other developers will modify your code
- Make sure those modifications can be tested
  ⇒ make your own tests available to others
  - Automation
  - Standardize on test framework
  - Standardize on how tests are run
  - Document your test approach

Focus on saving time

- No need to focus on coverage
  - “Smoke test” for all features
  - (Optional) built-in self-test of the complete firmware
  - Unit test for the feature being worked on
- Focus on tricky parts of implementation
- Put support for tests into the implementation
  - Assertions
  - Tracing
- Tests must run fast
  - So not complete!

Testing embedded software is special

- Software depends on hardware
- Limited access to hardware and hardware itself is limited
- Time is important
- Updates are essential

Embedded software is written for specific hardware

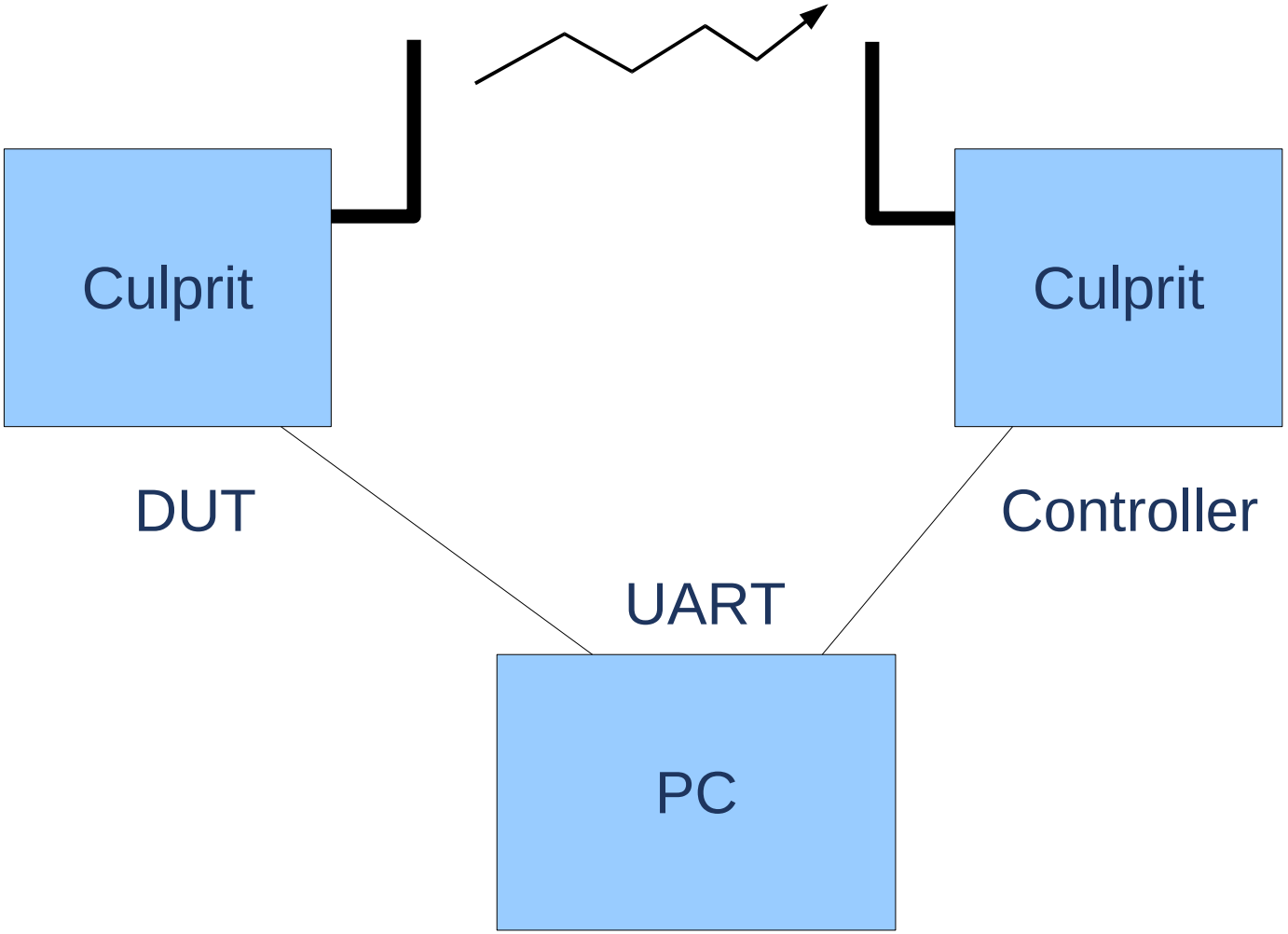
- Requires specific inputs and outputs
- Target has different architecture than PC
  - Endianness
  - Memory model
  - Hardware accelerators
- Target has limited resources
  - Memory
  - Disk
  - Speed

Hardware test setup

Make hardware available remotely

- Accessible over network
- I/O's can be controlled remotely
- Power can be controlled remotely

Example test setup for wireless device



Simulation overcomes limited access to hardware

- Different levels of simulation
  - Emulation: qemu (PowerPC, ARM, MIPS, M68K, SPARC)
  - Virtualization: KVM, VirtualBox
  - Stubbing/hardware abstraction
- PC has much more resources and performance
- Many more test tools on PC than on real platform
- Simulation of inputs is essential for reproducability
- Stubbing makes for the easiest testing and debugging
  - More effort to maintain the HAL
  - HAL isn't tested

In embedded software time is a factor

- Time is essential part of functionality
- Race conditions are time-dependent
- Test code (tracing, assertions) may affect timing

Testing must take into account timing

- Time-sensitive tests on target platform
  - Using file input
- Simulate time
  - Using profiling info to insert simulation delays
  - (Idea of Johan Cockx, Imec)

Working update system is essential for embedded systems

- If update goes wrong, device is dead
  - No alternative boot methods
  - Not reachable
- Developer must make sure that updates never fail
  - Power failure: corrupted software or filesystem
  - Integrity of transfer: corrupted software
  - Compatibility between pieces: e.g. kernel – module
  - Compatibility with hardware: e.g. wrong board support
- Package manager helps a lot, but no silver bullet
- Fallback boot should always exist

Testing and open source software

- Open source tools to support testing
  - Unit tests
  - Doctest
  - D-Bus

- Testing of open source tools
  - Gstreamer
  - Linux kernel

Not many open source tools exist to help developer with testing

opensourcetesting.org lists 36 unit test frameworks

But these are not entirely useful for the developer

- No need to have extensive reporting
- Fixtures and datasets don't give so much added value

Still some advantages

- Lowers threshold to add tests
- Validation team will love you

QtTestLib is a useful unit test framework

QtTestLib offers

- Selective execution
- Fixtures
- Data sets
- Benchmarking (= top-level profiling)
- Mock for user input in GUI

Python's documentation test is perfect for developer tests

Python documentation tests are in the code itself

- Low threshold to add test
- Easy to update test, it's right there
- Can easily run tests associated with specific function even if they call other functions

Unfortunately, nothing similar exists outside python (AFAIK)

D-Bus helps for testing

- D-Bus is great for IPC
- It is also a great tool for testing
  - Bindings for scripting language
  - Create stub implementation of required methods/signals
  - Verify method return value against expected return value
  - Insert signals into the code to expose internals

Developer tests are even more important for open source

- There is no validation team
  ⇒ cycles are longer
- Contributors don't know the code
  ⇒ higher risk of breaking things
- Many contributors
  ⇒ larger benefit from sharing tests
- Contributors only interested in added feature
  ⇒ needs to be motivated to also update tests

Gstreamer has testing but it's not good enough

- All Gstreamer elements have a test required to enter gst-plugins-good
- Many stub elements e.g. videotestsrc

However:

- Not trivial to run specific test
- May take long (e.g. videocrop)

Gstreamer should:

- Add boilerplate to test more than just buffer I/O e.g. handling of QoS, caps, events
- Put the tests closer to source code, so contributors see them
- Split into fast individual tests

Linux kernel has stubs and test framework

- For most device types, stub implementation exists
  - For USB gadget: dummy_hcd, zero
  - For MTD (= flash): block2mtd

- Linux Test Project (LTP) offers a large test suite
  - Mainly for filesystems and networking
  - Mainly regression, load and performance tests
  - Simple to select specific test
  - Good boilerplate ⇒ good base to write new test

Conclusions

- Developer should write tests from the start
  It saves time!
- Share the tests with other developers
- Even more important for open source projects
- For embedded systems,
  hardware abstraction / stubs are essential

www.mind.be

www.essensium.com

Essensium NV
Mind - Embedded Software Division
Gaston Geenslaan 9, B-3001 Leuven
Tel : +32 16-28 65 00
Fax : +32 16-28 65 01
email : info@essensium.com

Spread the word !

- Feel free to use these slides

  elinux.org, look for ELC Europe 2010

- Text at

  http://mind.be/?page=embedded-software-testing

  http://mindlinux.wordpress.com