ALMA Common Software

Basic Track

Test Driven Development
Unit testing and TAT

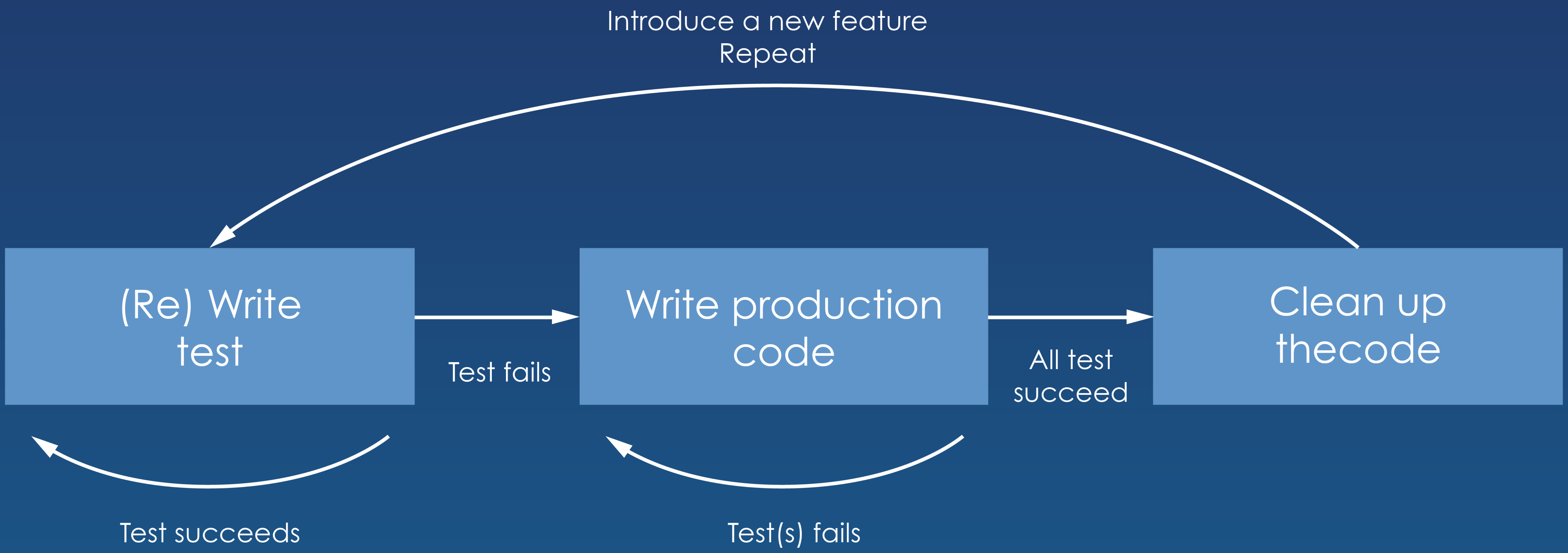
What is it?

In a nutshell

✧ Development process that relies on very short development cycles

✧ Tests written before functionality

Workflow



Introduce a new feature
Repeat

(Re) Write test

Test fails

Write production code

All test succeed

Clean up thecode

Test succeeds

Test(s) fails

How to test?

- Unit testing
    - Atomic code functionality tests
- xUnit code-driven testing framework:
    - CppUnit
    - jUnit
    - PyUnit

What to test? What not to test?

For sure:

- ✧ Interfaces
- ✧ Critical and/or non trivial portions of the code

Avoid testing:

- ✧ Code without added value, such as getters/setters and delegations

Be cautious:

- ✧ Over-testing is a waste of time

xUnit

- Test case: basic unit
- Test suite: set of tests that share the same test conditions
    - setup()
    - tearDown()
- Assertions

ACS TAT

- Automated test driver
- Initializes environment (ACS)
- Pass criteria based on comparing test output against an output template
- Can also run other kind of tests besides xUnit-based ones

Integration Tests

- ✧ Tests the interactions among software modules
- ✧ Tests deployment configuration
- ✧ Integration CDB
- ✧ Hardware configuration, if necessary
- ✧ Individual module unit tests must pass to integrate

Development team

Responsible for:

- ✧ Adding (at least one!) unit test(s) for your component
- ✧ Prepare a test environment
- ✧ Run the test(s) until the work
- ✧ Once tests pass, check in all your code, test code and relevant configurations (TAT files, CDB) to the project configuration server
- ✧ Document your work right away and in parallel with the development!

Advise: When done, take a well deserve coffee break…

Integration and testing team

Responsible for:

- ✧ Continuous (and automatic) integration
- ✧ Quality control during development
- ✧ Build every time changes are detected
- ✧ Identifying compilation and failures against regression tests
- ✧ Facilitating the parties responsible for the failure and more importantly the parties responsible for fixing it
- ✧ Revert to the last working checkpoint in case of a failure

Advise: Use automated tools like Jenkins, the integration team key tool

Questions?

Acknowledgements
ACS presentations were originally developed by the ALMA Common Software development team and has been used in many instances of training courses since 2004. Main contributors are (listed in alphabetical order): Jorge Avarias, Alessandro Caproni, Gianluca Chiozzi, Jorge Ibsen, Thomas Jürgens, Matias Mora, Joseph Schwarz, Heiko Sommer.

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.