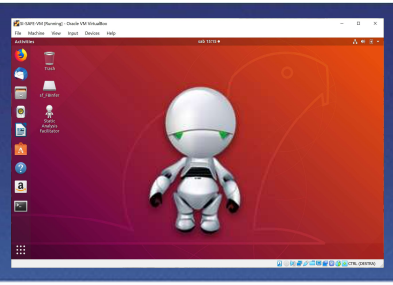
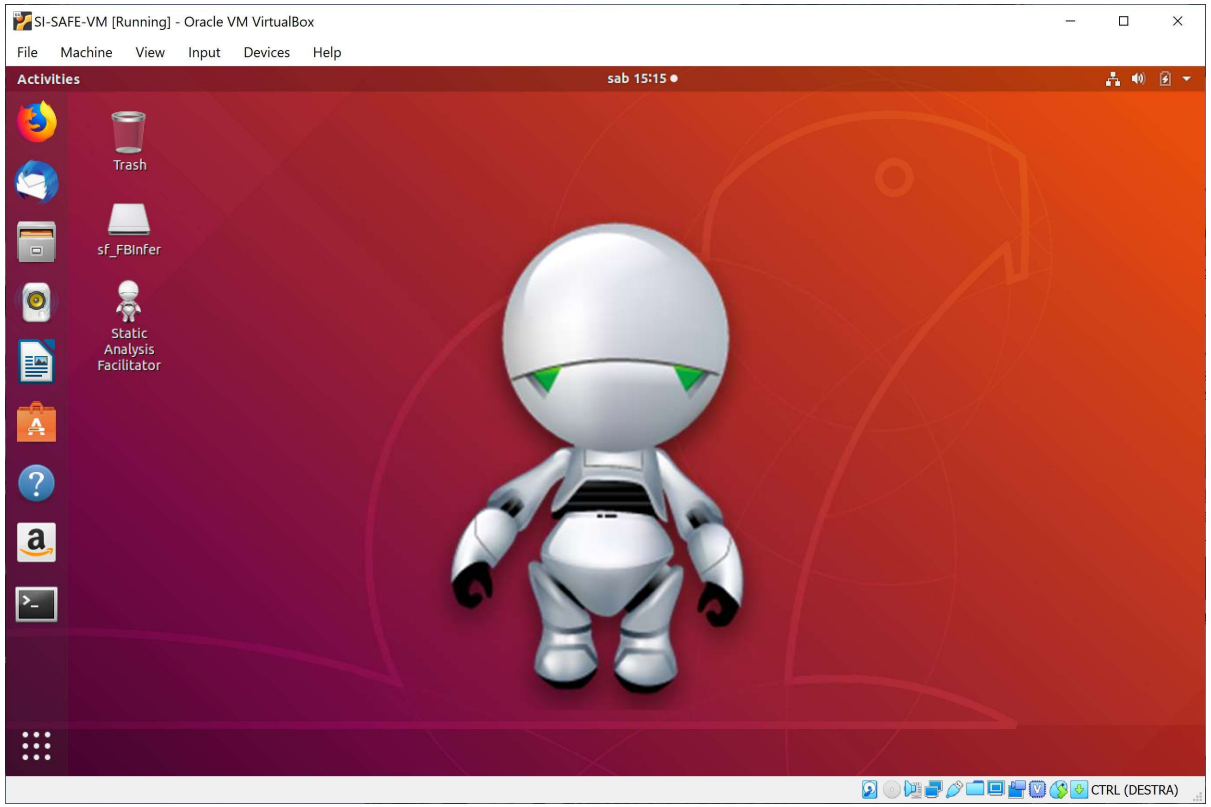


SPAZIO IT

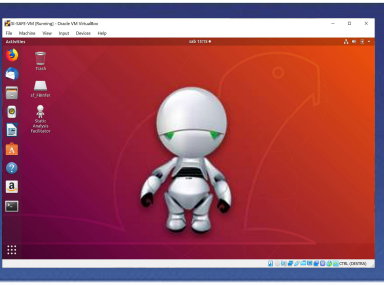
SAFE TOOLSET



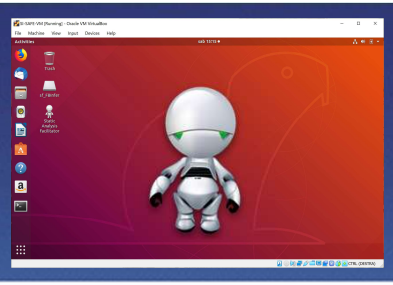
Maurizio Martignano
Spazio IT – Soluzioni Informatiche s.a.s
Via Manzoni 40
46030 San Giorgio Bigarello, Mantova
https://www.spazioit.com

December 2019

Agenda

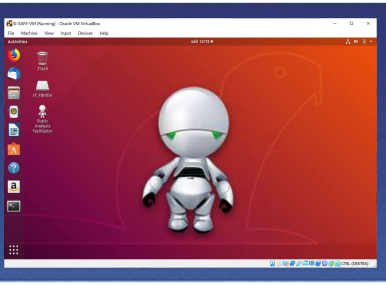


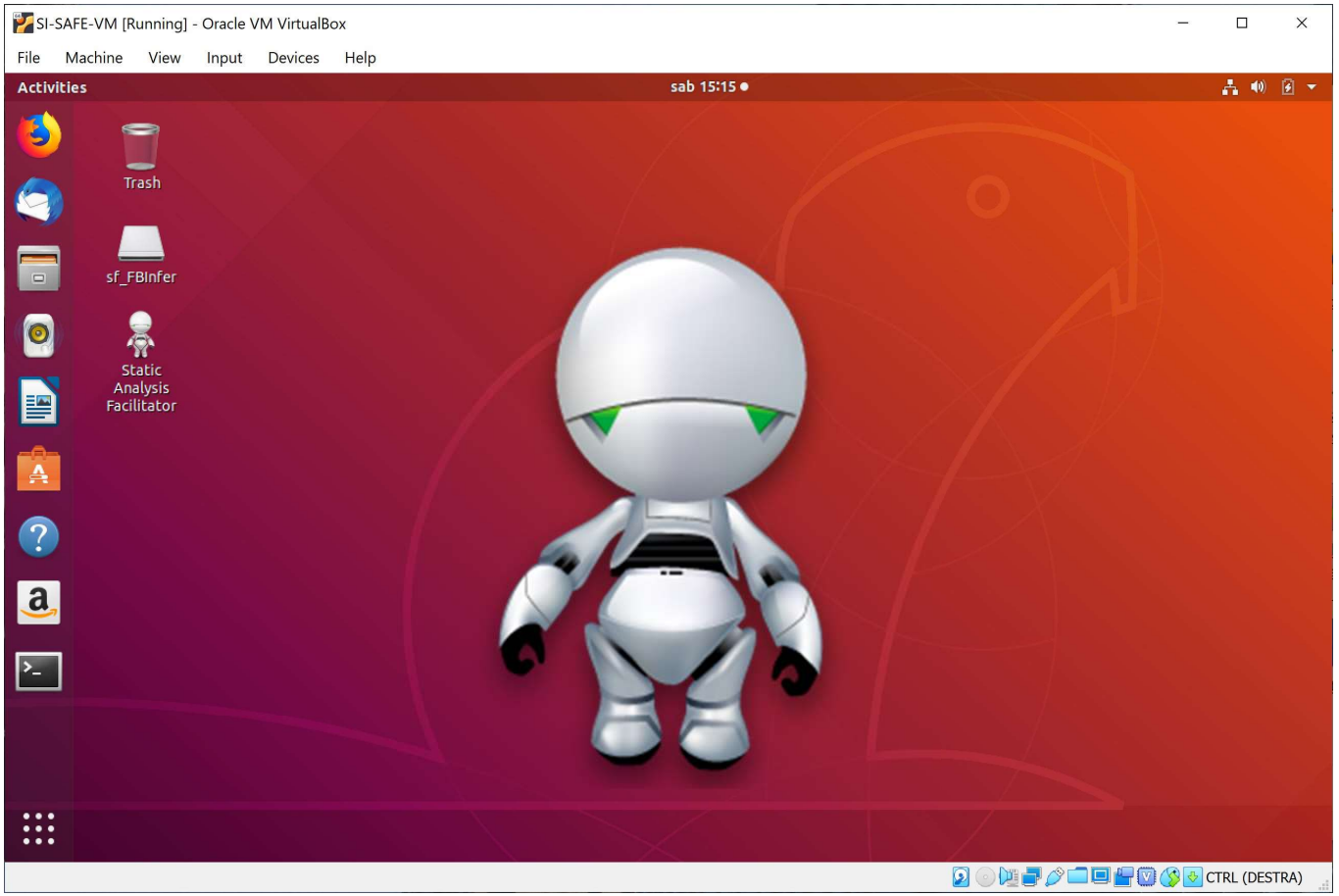
Agenda



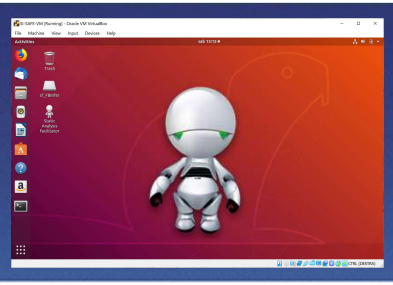
- SAFe Toolset
- Toolset Workflows
- The SAFacilitator
- An example: Crazyflie
- SAFe Toolset Online Help
- Future Evolution

SAFe Toolset



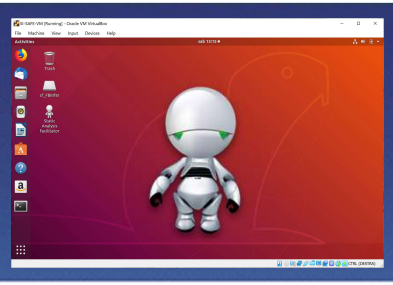


SAFe Toolset



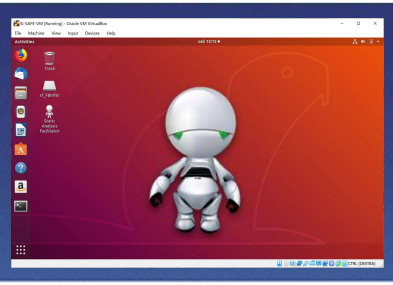
- The **SAFe** (Static Analysis Framework) **Toolset** is a set of **open source tools**, packaged in easily reusable form (currently, an Ubuntu Virtual Machine), that can be used to perform **Software Verification** and **Validation**.
- A set of analyzers for C/C++:
  - **cppcheck** – v. 1.87 - http://cppcheck.sourceforge.net/ - a C/C++ static analyzer.
  - **Clang** – v. 9.0.0 - https://clang.llvm.org – the "new" compiler toolset from LLVM Foundation, with its **Clang-SA** and **Clang-Tidy** static analyzers.
  - **SonarQube** – v. 8.0. – https://www.sonarqube.org/ - a code quality platform used to show and manage the issues found by the static analyzers.

SAFe Toolset



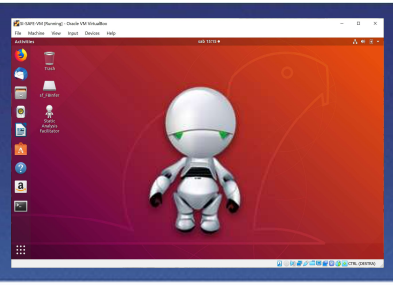
- A set of analyzers for Java:
  - **SonarJava** – version 5.4.1 - https://www.sonarsource.com/products/codeanalyzers/sonarjava.html from SonarSource ;
  - **SpotBugs** – v. 3.1.12 - https://spotbugs.readthedocs.io/en/stable/ - was **FindBugs**;
  - **Checkstyle** – v. 8.22 – https://checkstyle.sourceforge.io/ – to check compliance to formatting standards;
  - **JDepend** – v. 1.1.1 – https://github.com/willemsrb/sonar-jdepend-plugin – to verify architectural / design dependencies;
  - **Jacoco** – v. 0.8.4. – https://www.jacoco.org/ – to execute coverage analysis.

SAFe Toolset



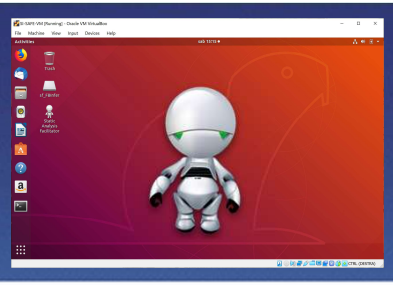
- Optionally the SAFe VM may also contain:
  - **PC-Lint** (or PC-Lint Plus) – v. 9.0.0L - https://www.gimpel.com/ - but its license needs to be acquired from Gimpel.
- Apart from the static analyzers the SAFe VM contains also some (native and cross) build environments, that is:
  - **GNU GCC** Version 7.3.0 - https://gcc.gnu.org/gcc-7/ - Native
  - **Clang** Version 9.0.0 - - https://clang.llvm.org - Native and Cross (Multiplatforms – use the command "llc --version" to see the supported architectures).
  - **BCC2**: Bare-C Cross-Compiler System for LEON2/3/4 GCC 7.2.0 - https://www.gaisler.com/ - Cross.
  - **GNU Arm Embedded Toolchain** - v. 5-2016-q3 - https://launchpad.net/gcc-arm-embedded - Cross.
  - OpenJDK – v. 12.0.1. – http://openjdk.java.net/projects/jdk/12/ – for Java

SAFe Toolset



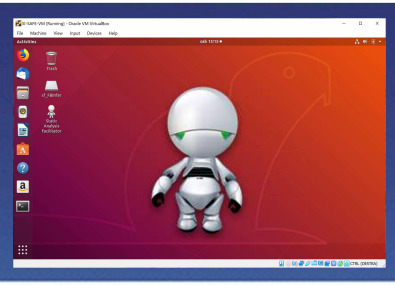
- Should a user need to work on a codebase not supported by the provided build environments, she would need to install the corresponding compilation toolchain.
- Additionally Spazio IT has complemented the SAFe Toolset with:
  - a specially **modified version of SonarQube** - https://www.sonarqube.org/ ;
  - a specially **modified version of the SonarQube C++ Community Plugin** - https://github.com/SonarOpenCommunity/sonar-cxx ;
  - the **SAFacilitator** – an application largely simplifying the static analyzers usage and the integration of their results into SonarQube –more info @ https://www.spazioit.com/pages_en/sol_inf_en/code_quality_en/safe-toolset/

SAFe Toolset

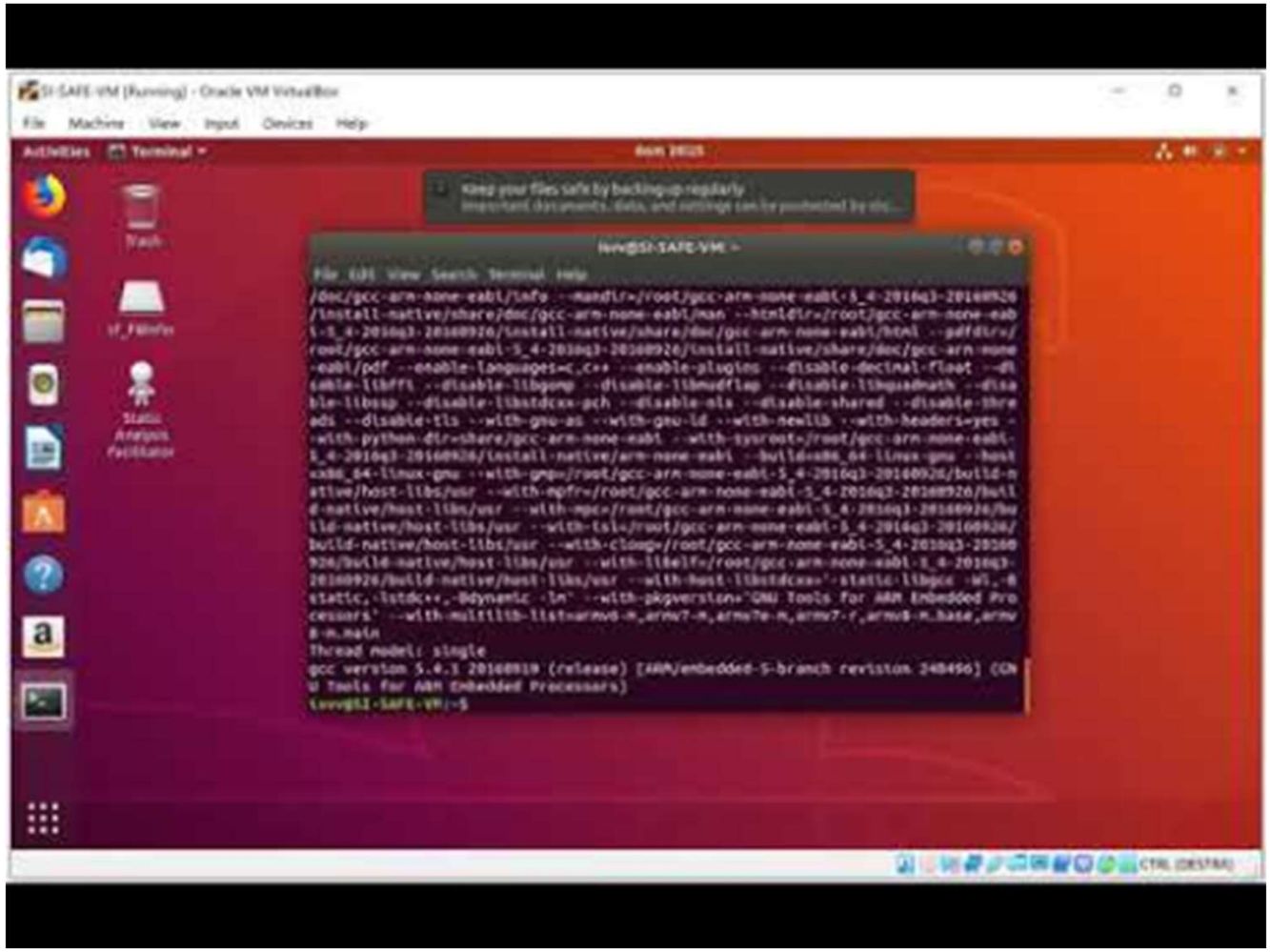


- The development of the SAFe Toolset has been funded by the European Space Agency Contract # RFP/3-15558/18/NL/FE/as.

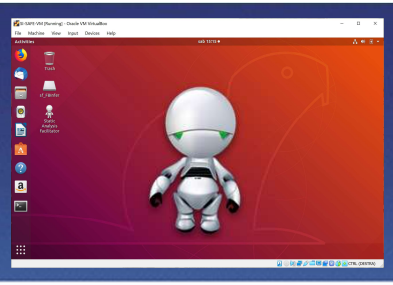
SAFe Toolset



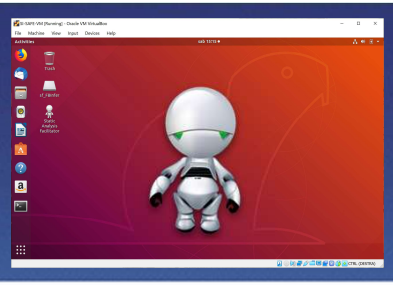
- A tour in the actual SAFe VM.

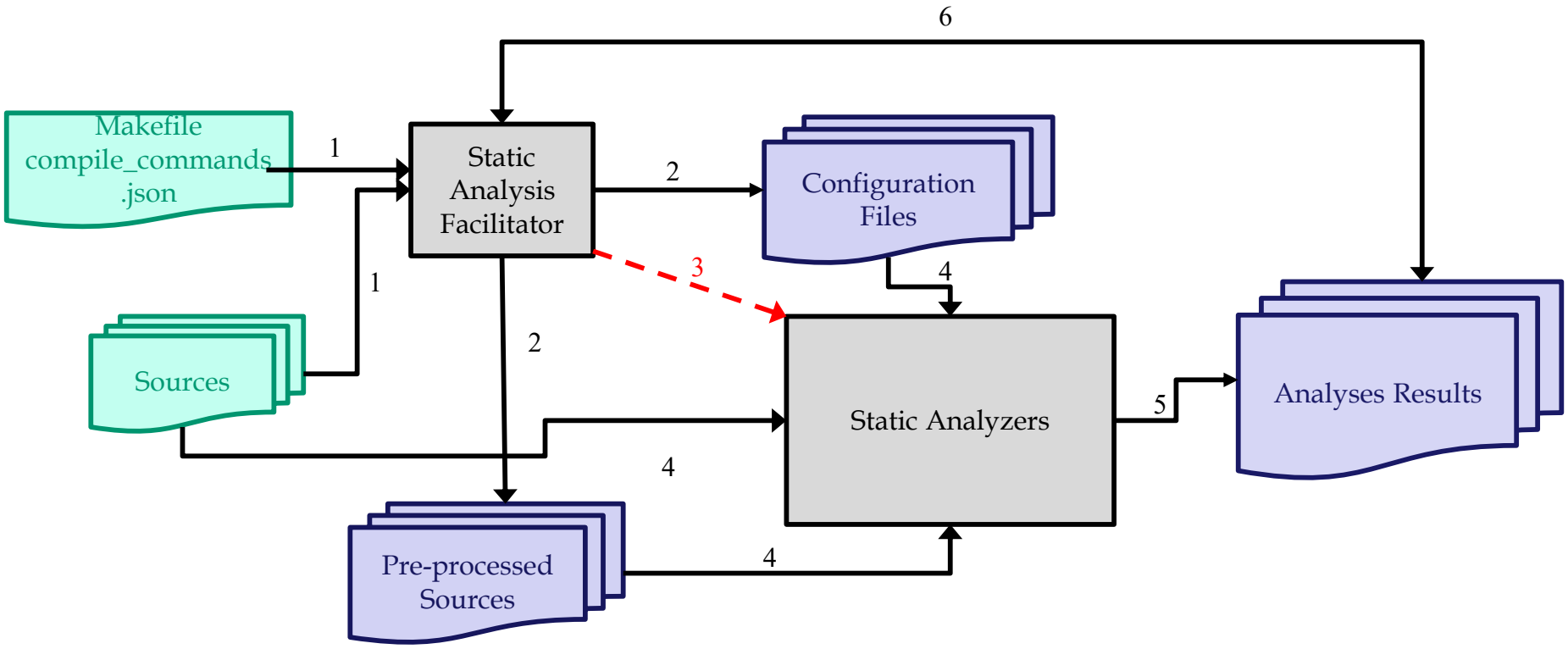


SAFe Toolset Wokflows

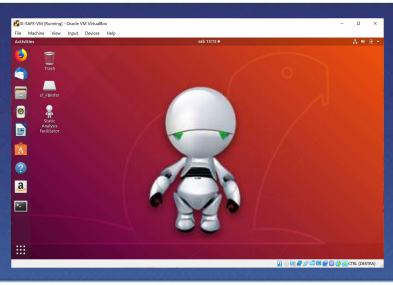


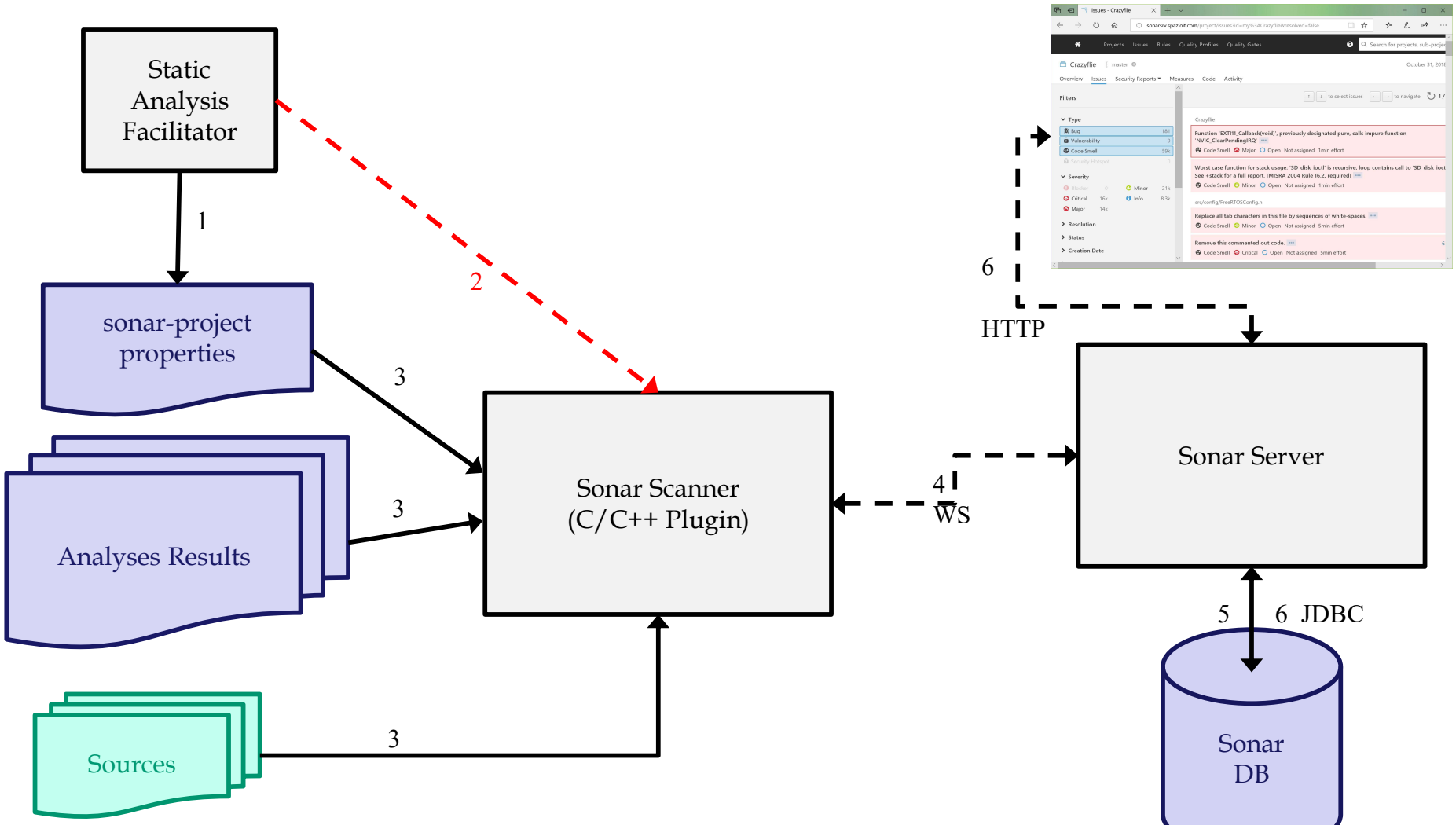
Toolset Workflow (C/C++)



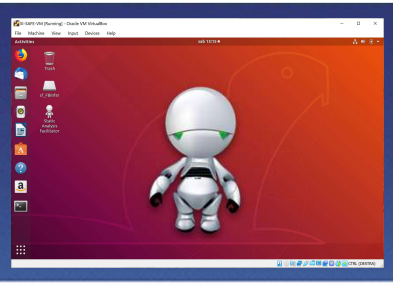


Toolset Workflow (C/C++)



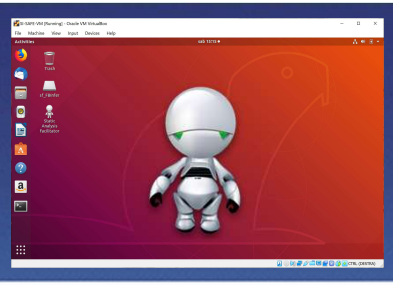


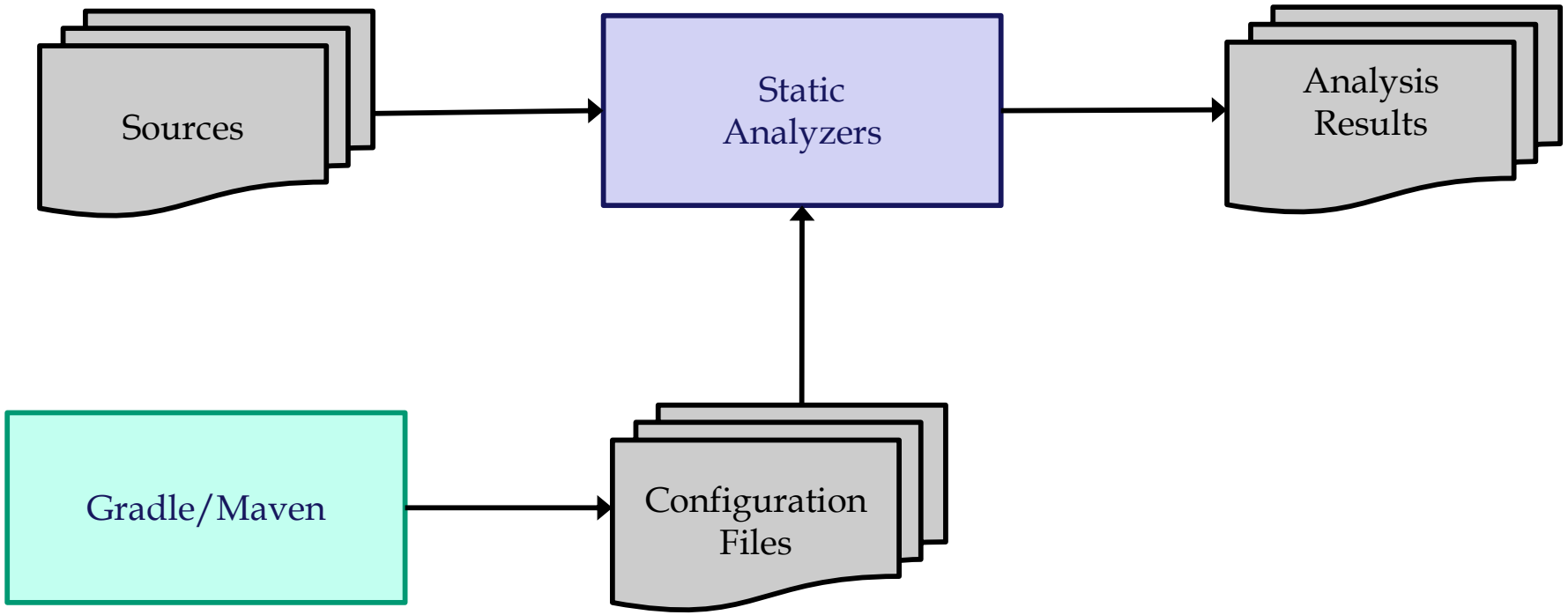
Toolset Workflows: C/C++ Workflow



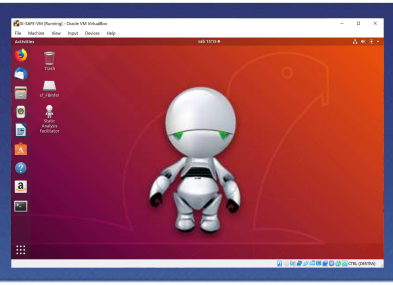
- Get the codebase, Compile it and Obtain the Compilation Database
- Normalize the Compilation Database
- Generate the Static Analyzers Configuration Files according to your needs
- Run the Static Analyzers
- Configure SonarQube Analyses
- Gather Static Analyzers Results into SonarQube
- Review Results

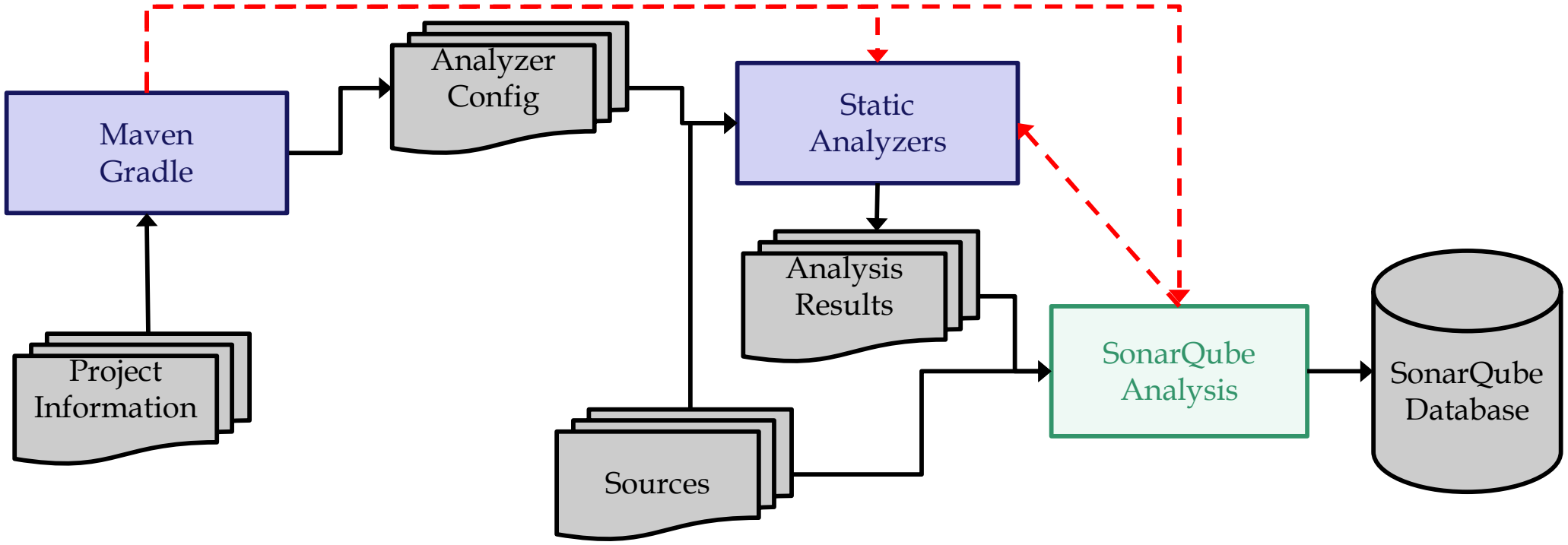
Toolset Workflow (Java)



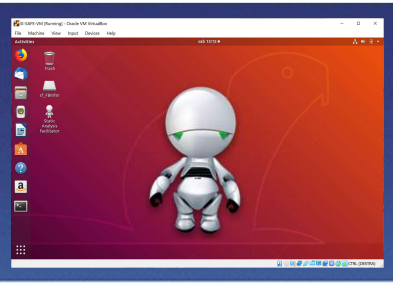


Toolset Workflow (Java)



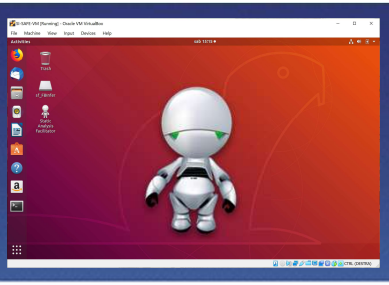


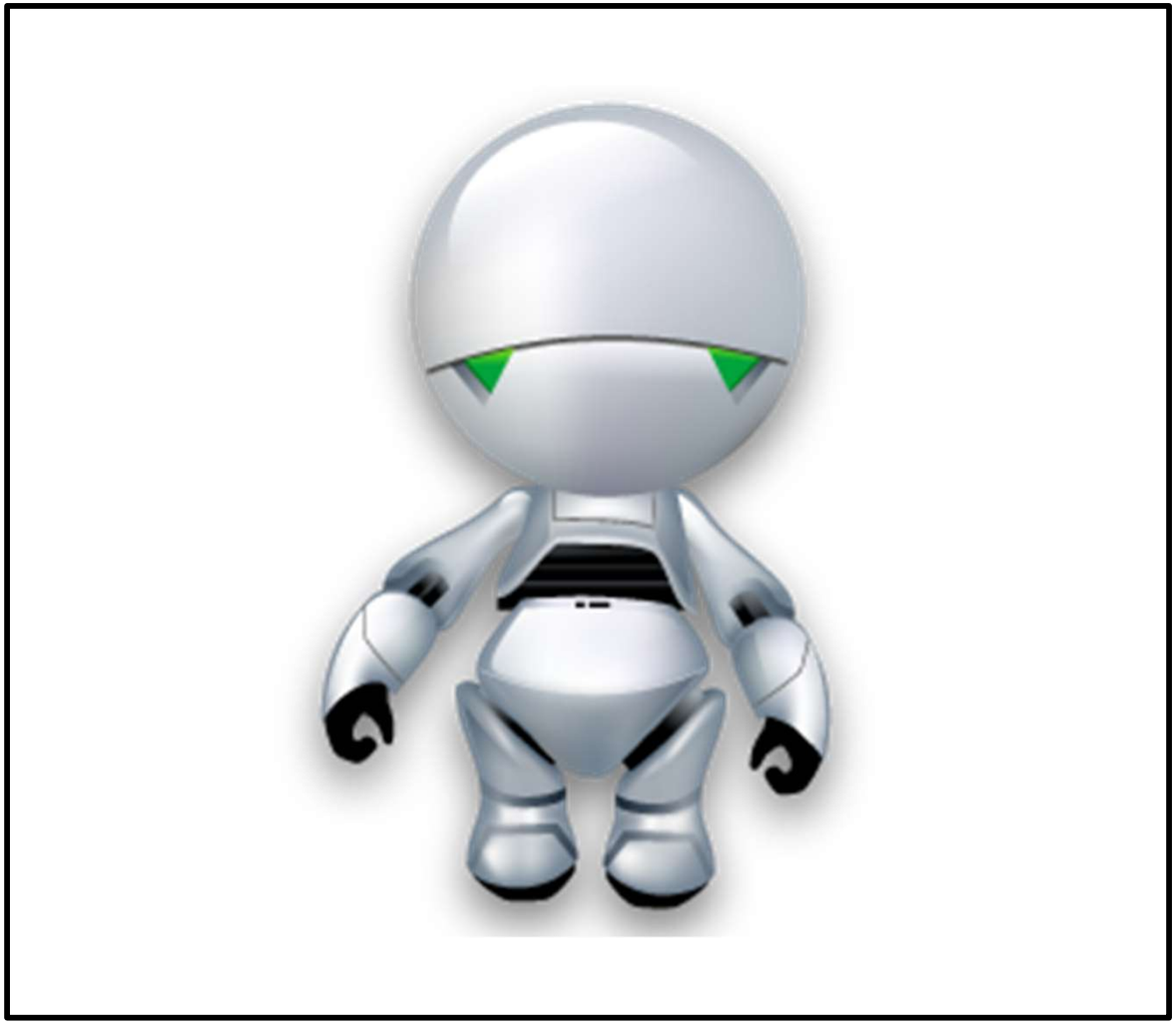
Toolset Workflows: Java Workflow



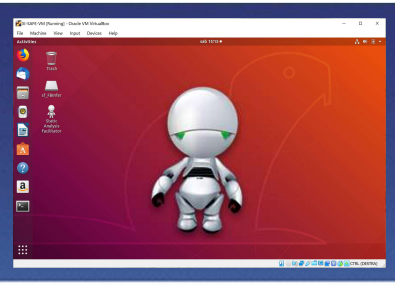
- Get the codebase
- Activate Integration between build tool (Gradle/Maven) and SonarQube
- Configure SonarQube Analyses
- Run analyses
- Review Results

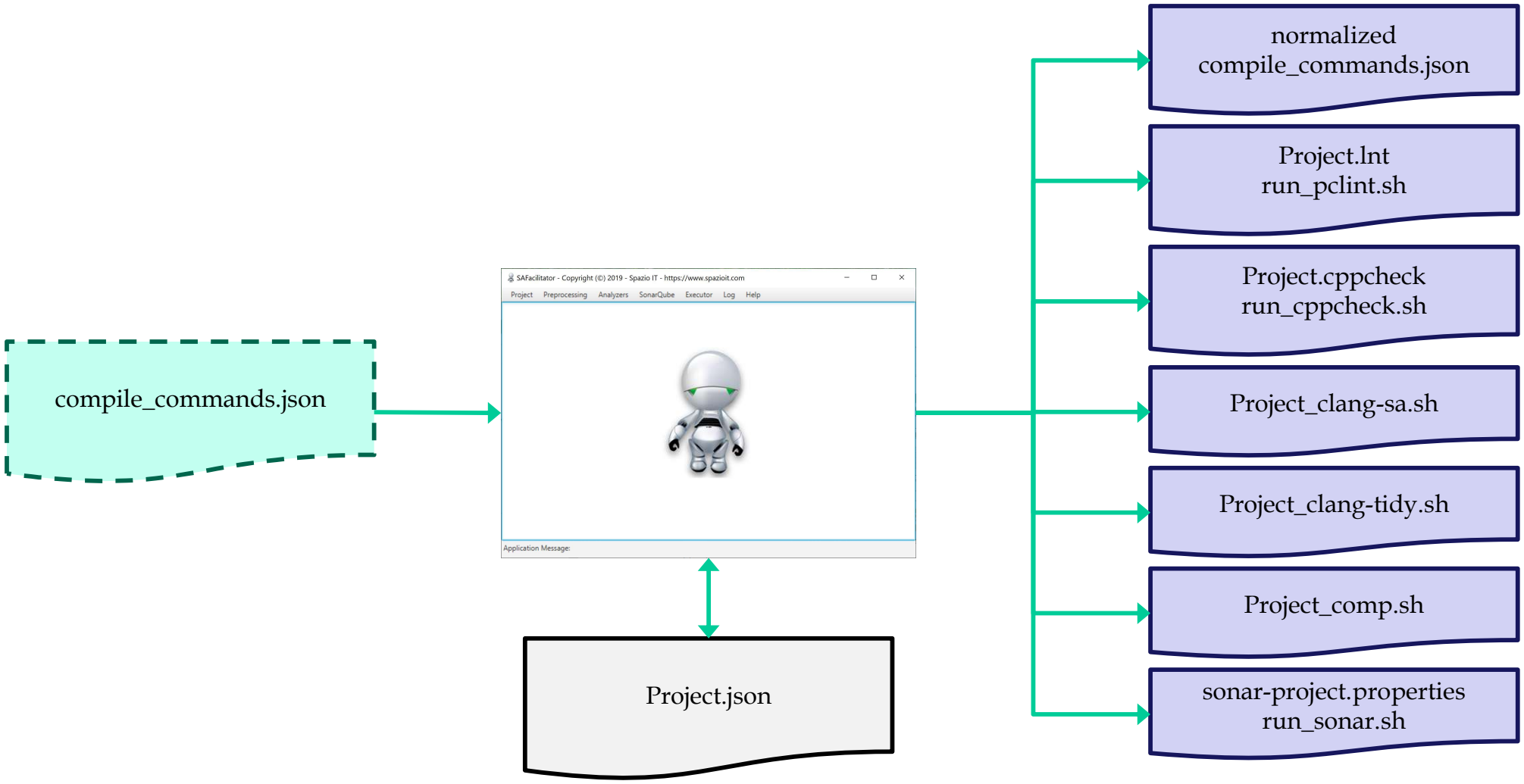
The SAFacilitator



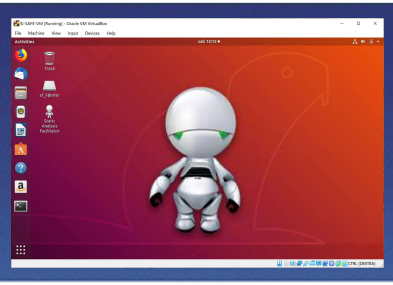


The SAFacilitator

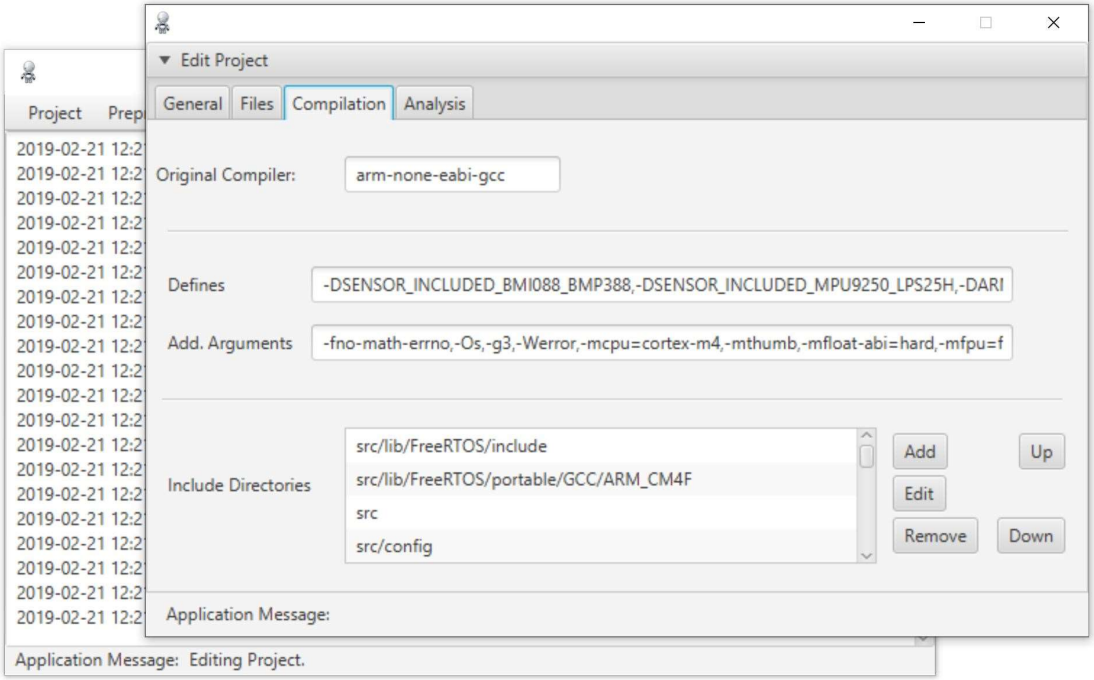




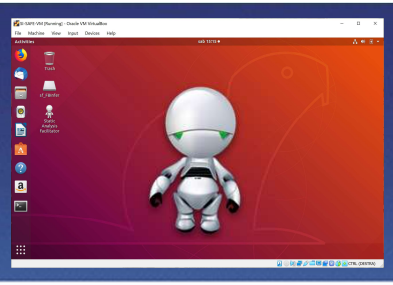
SAFacilitator dual nature



- SAFacilitator is not only a (Java FX) GUI Application…



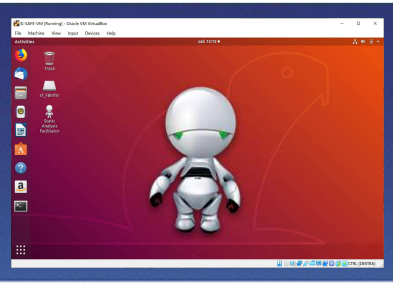
SAFacilitator dual nature



- … it is also a Console Application that can be called by scripts.

```
Windows PowerShell
PS C:\NetBeansProjects\SAFacilitator\dist> java -jar .\SAFacilitator.jar -l C:\Crazyflie\Crazyflie.js
on -cc C:\Crazyflie\compile_commands.json -pa -ea -ppa
2019-02-21 13:16:02 - Preparing PC-Lint...
2019-02-21 13:16:02 - PC-Lint prepared.
2019-02-21 13:16:02 - Preparing Cppcheck...
2019-02-21 13:16:02 - Cppcheck prepared.
2019-02-21 13:16:02 - Preparing Clang-SA...
2019-02-21 13:16:02 - Clang-SA prepared.
2019-02-21 13:16:02 - Preparing Clang-Tidy...
2019-02-21 13:16:02 - Clang-Tidy prepared.
2019-02-21 13:16:02 - Preparing GCC...
2019-02-21 13:16:02 - GCC prepared.
2019-02-21 13:16:02 - Executing PC-Lint...
2019-02-21 13:16:02 - PC-Lint executed.
2019-02-21 13:16:02 - Executing Cppcheck...
2019-02-21 13:16:02 - Cppcheck executed.
2019-02-21 13:16:02 - Executing Clang-SA...
2019-02-21 13:16:02 - Clang-SA executed.
2019-02-21 13:16:02 - Executing Clang-Tidy...
2019-02-21 13:16:02 - Clang-Tidy executed.
2019-02-21 13:16:02 - Executing GCC...
2019-02-21 13:16:02 - GCC executed.
```

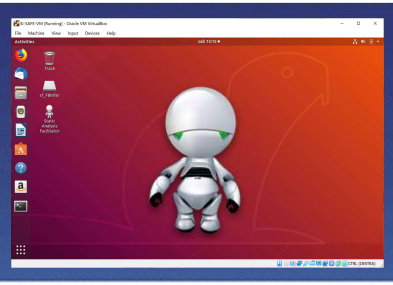
SAFacilitator dual nature



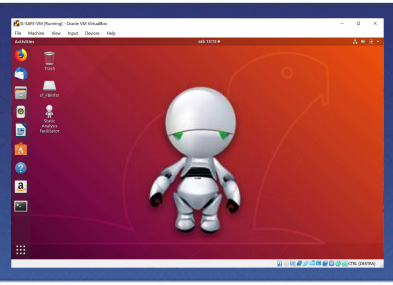
- The idea is to use SAFacilitator as GUI Application at the beginning of a SVV Project - to configure it and set it up properly -

- and then to use the tool as an utility to be called during recurring operations – to run analyses, save the data into SonarQube, etc…

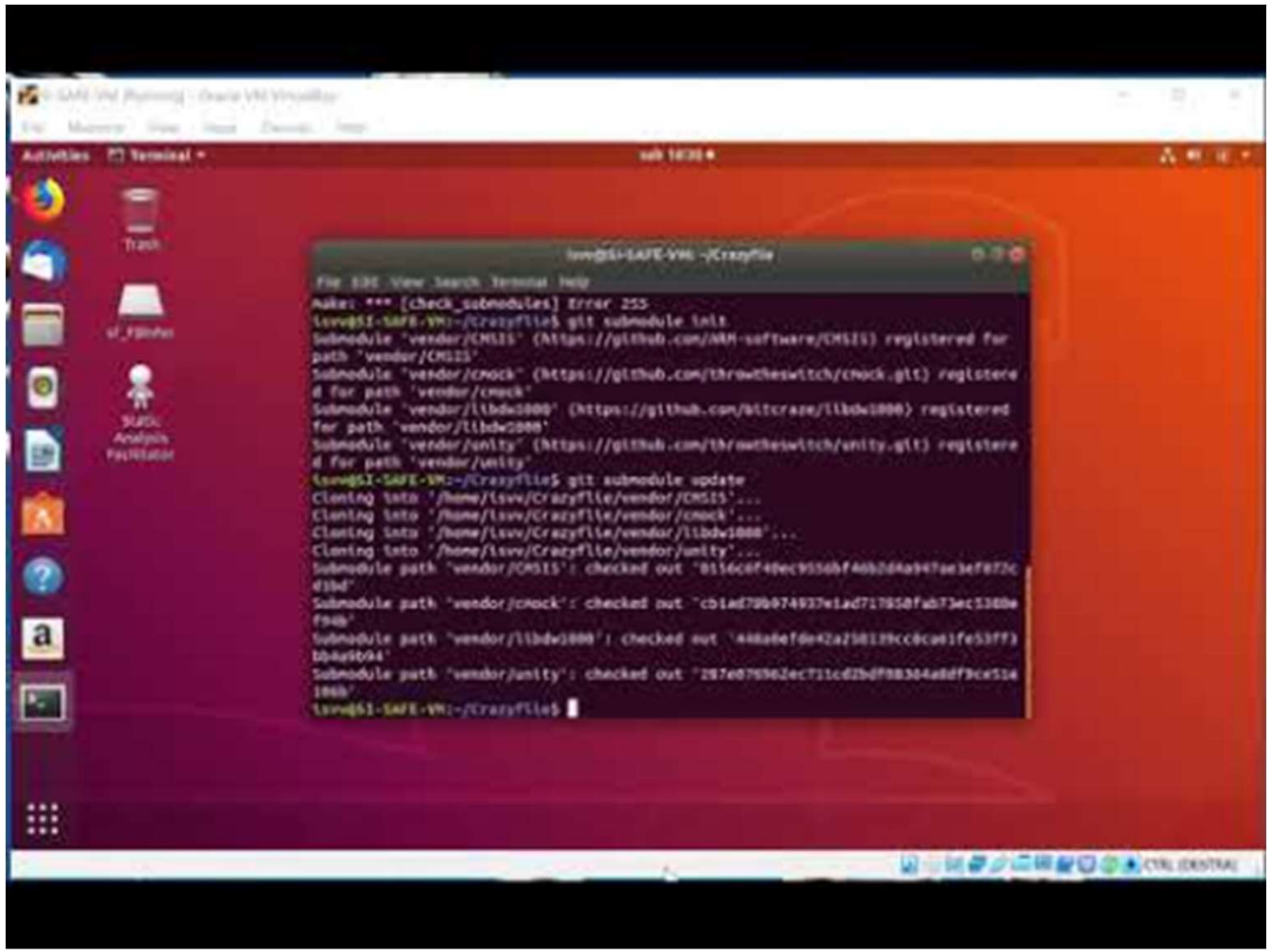
An example: Crazyflie 2.1



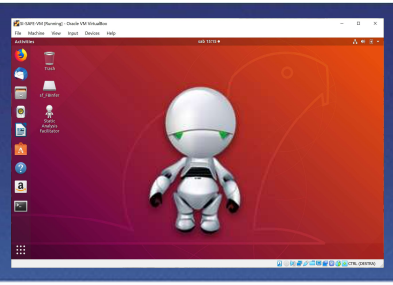
Step 1



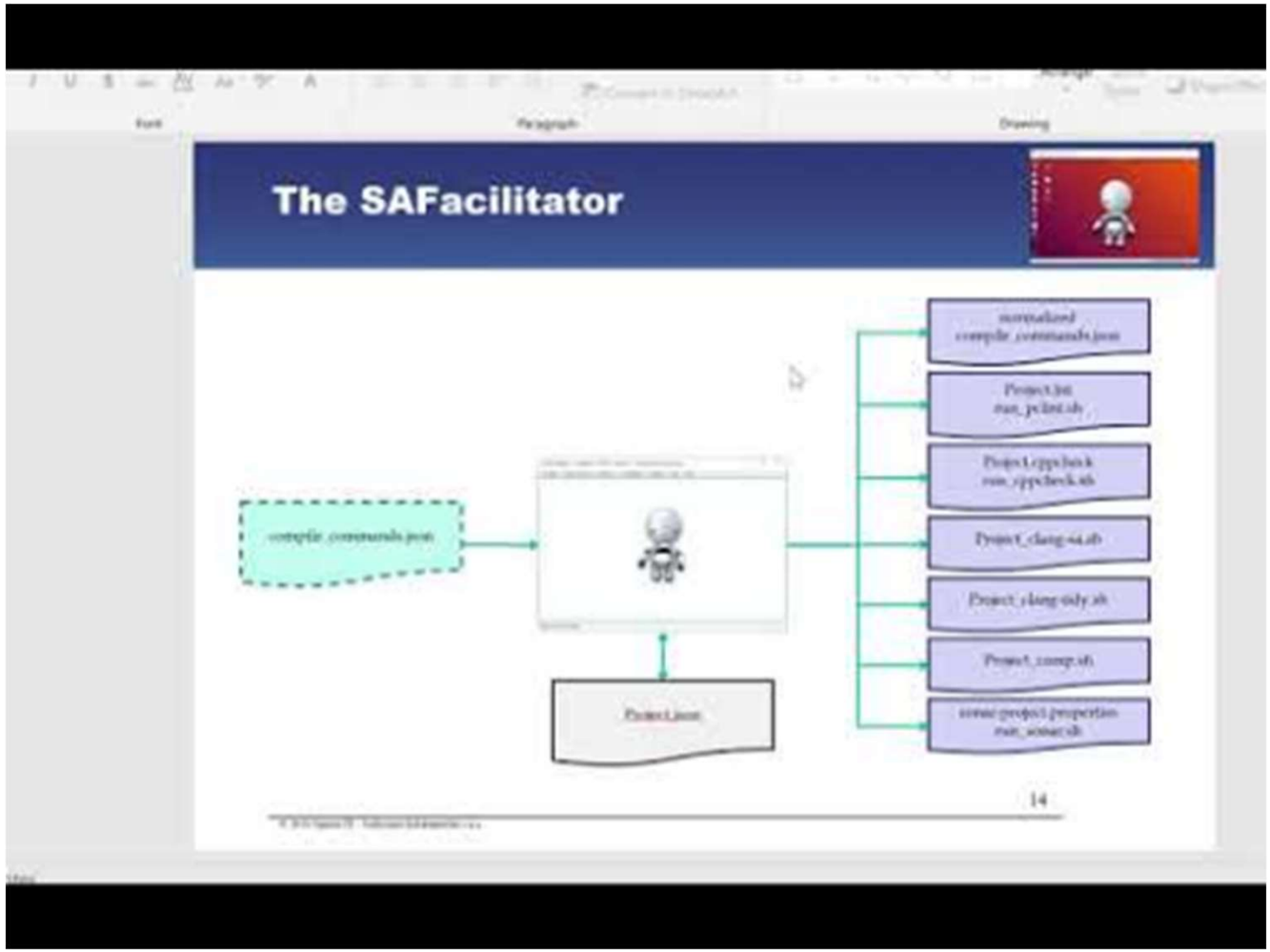
- Get the code - compile it - obtain the Compilation Database.



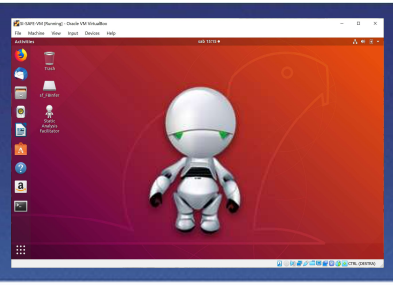
Step 2



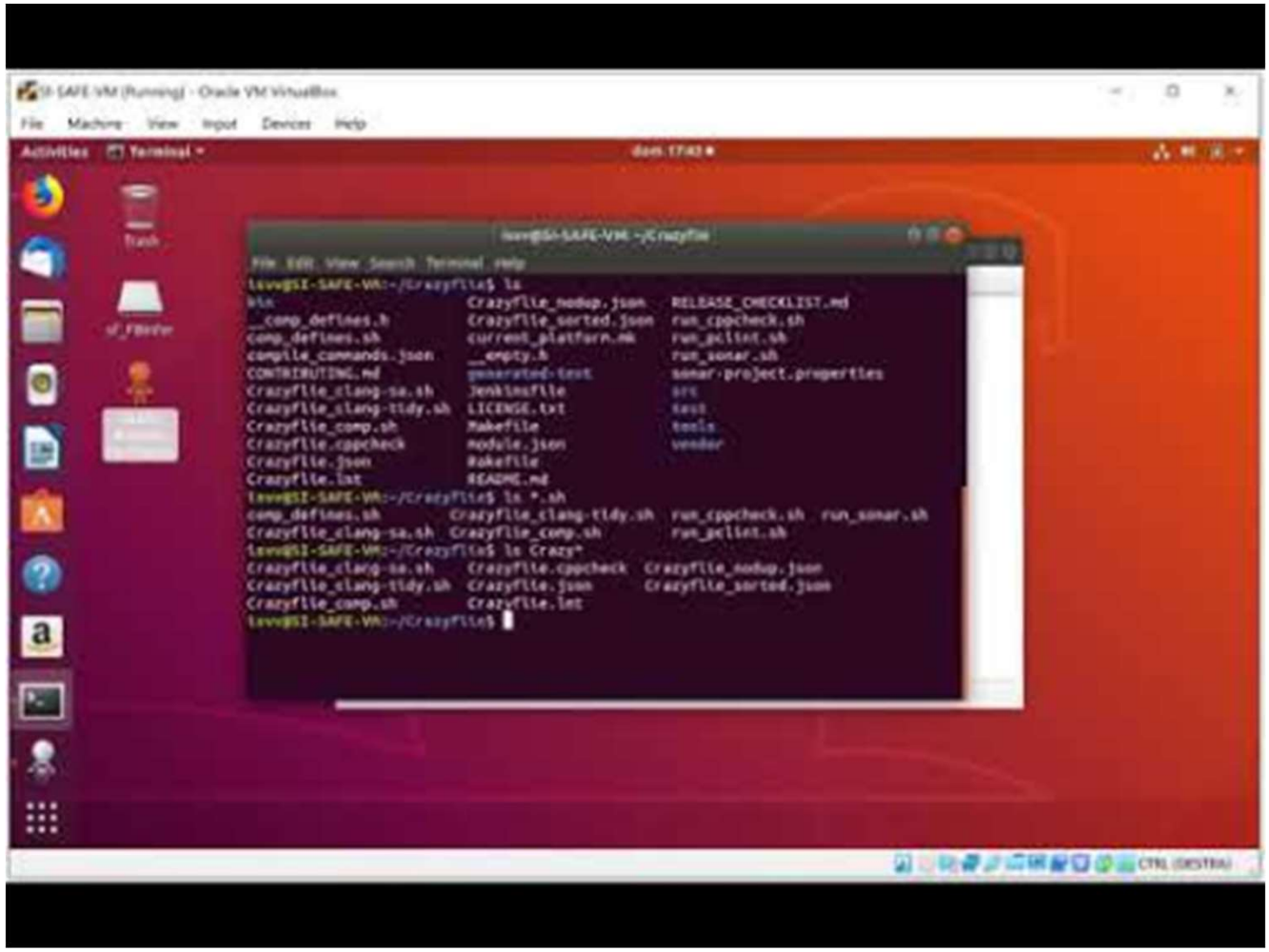
- Normalize the Compilation Database.



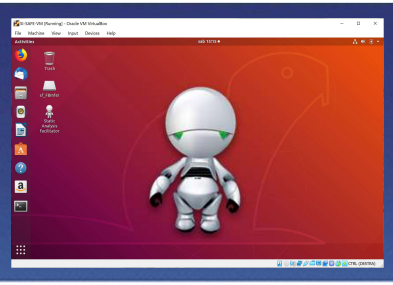
Step 3



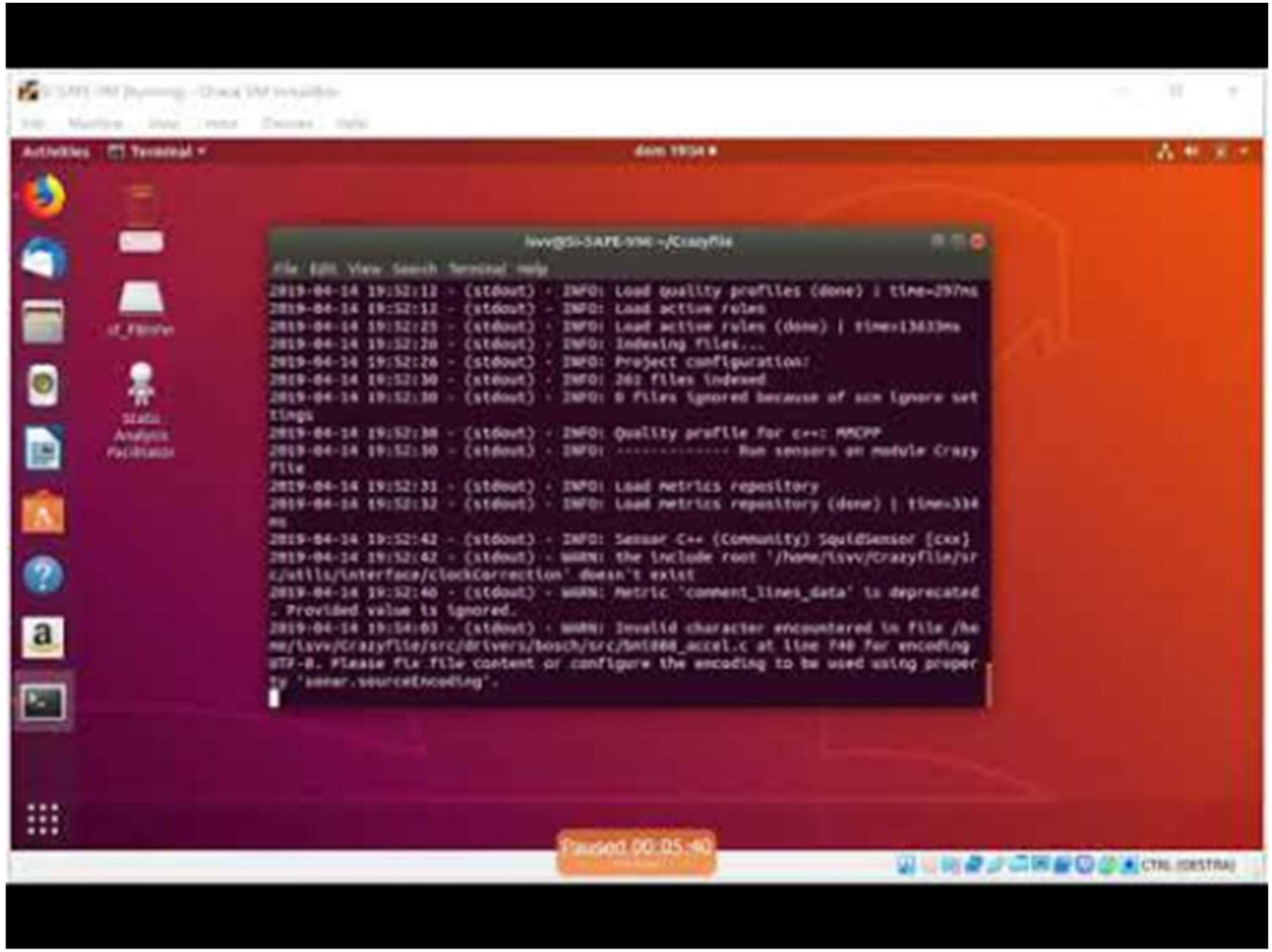
- Configure and run the Static Analyzers.



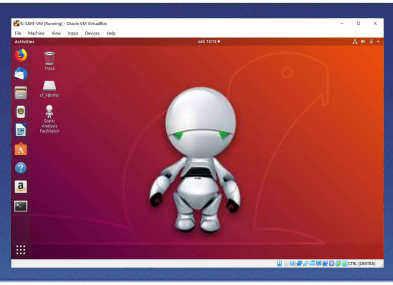
Step 4



- Gather the Analyses Results into SonarQube and Review Them.

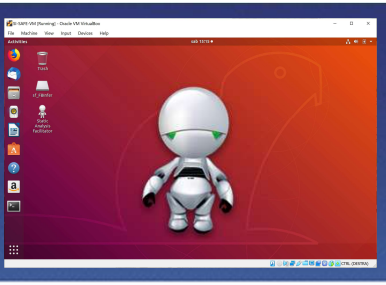


SAFe Toolset Online Help

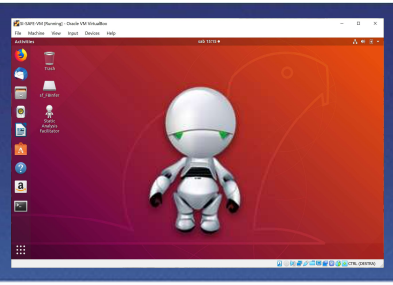


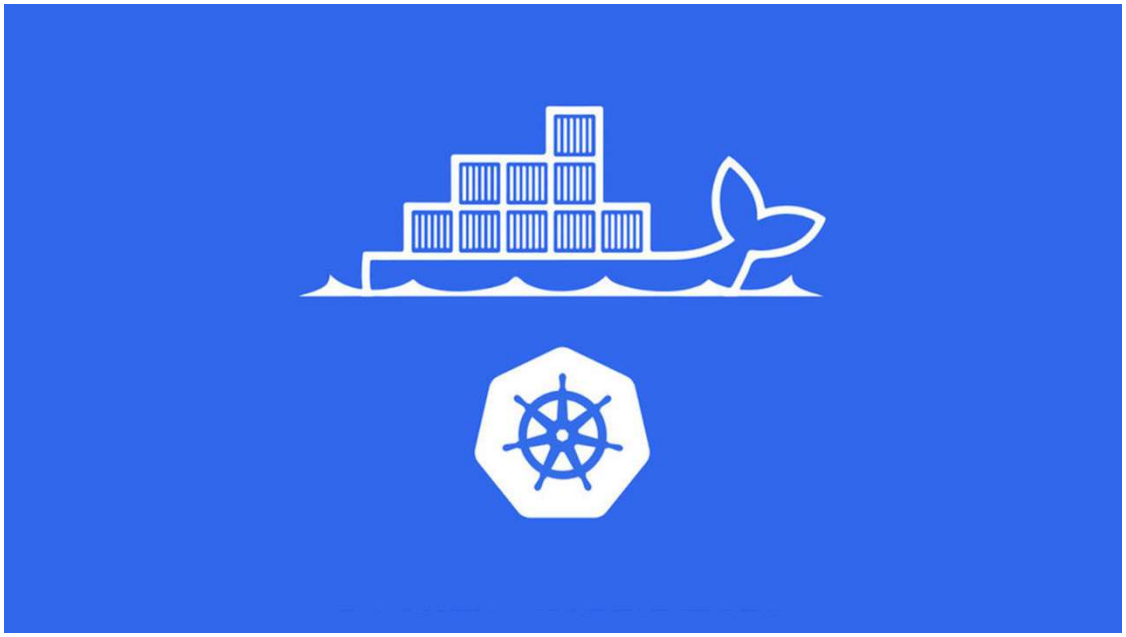
https://www.spazioit.com/SAFeToolsetHelp/

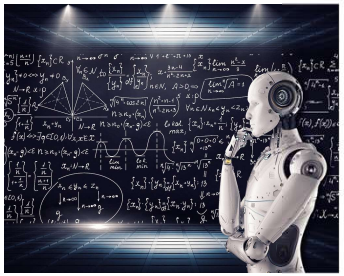
Future Evolutions?



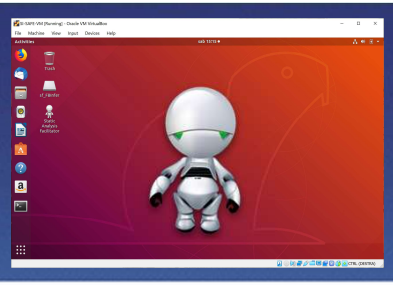
SAFe Toolset Future Evolutions







SAFe Toolset Webpage



Please visit the SAFe Toolset Webpage:

https://www.spazioit.com/pages_en/sol_inf_en/code_quality_en/safe-toolset-en/

Thank you for your time!