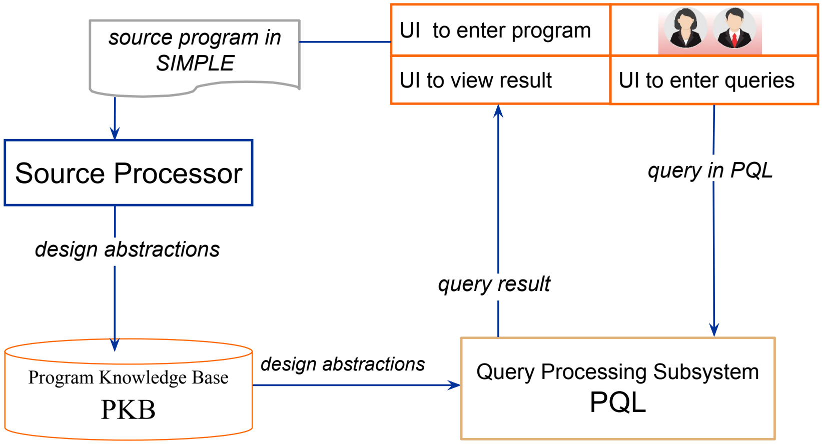
Tools for the Project

CS3203: Software Engineering Project

13 Aug 2021

Agenda

SPA - Static Program Analysis Tool



SPA - Static Program Analysis Tool

Summary of Environment/Tools

- **Project Management:** Github
- **Source Code Management:** Github
- **DevOps/CI:** Github Actions, Travis, AppVeyor, or others
- **Language:** C++17
- **Test Framework:** VS Unit Testing or Catch
- **SPA Solution:**
    - **spa-win**: Windows and VS Enterprise 2019
    - **spa-cp**: Cross-Platform
        - **Target Environment:** Windows, MacOS or Fedora Linux
        - **IDE:** VS Enterprise 2019, CLion, or others

SPA - Static Program Analysis Tool

Two Target Projects

- Autotester (Mandatory - Used in grading, impl. TestWrapper)
- GUI (Optional - For your own benefit, testing etc...)

```
// a default constructor
TestWrapper::TestWrapper() {
  // create any objects here as instance variables of this class
  // as well as any initialization required for your spa program
}

// method for parsing the SIMPLE source
void TestWrapper::parse(std::string filename) {
  // call your parser to do the parsing
  // ...rest of your code...
}

// method to evaluating a query
void TestWrapper::evaluate(std::string query, std::list<std::string>& results){
  // call your evaluator to evaluate the query here
  // ...code to evaluate query...

  // store the answers to the query in the results list (it is initially empty)
  // each result must be a string.
}
```

Version Control System - Git

Make sure you brush up on your git skills:

`https://try.github.io/`

Minimally, how to do the 3 things:

- **Git add:** Add files to be committed
- **Git commit:** Commit with message
- **Git push:** Push to remote Git server

Version Control System - Git

- All teams **must** use the provided Git repository provided under the Github org `https://github.com/nus-cs3203`
  - Do not fork, use branches.
  - Your tutors will be added, it will be monitored.
  - Git repository will be preloaded with default SPA solution.
- For repo to be created, each team must provide the following:
  - All GitHub IDs of the members of the team.
  - Choice of startup solution (Windows or Cross-platform).
- Please fill in the required infomation into README.md:
  - Target Environment - Include OS and Toolchain.
  - Any additional build instructions.
  - Team members, contact details and dev. OS/Toolchain

Project Management

All teams **must** use the provided Github team provided under the Github org `https://github.com/nus-cs3203`:

- Discussion on Github
- Issues
- Github Actions
- Wiki

Your tutors will be added, your Github group will be monitored.

Startup SPA Development

Windows Startup SPA Solution (Official/Recommended):

Easy

VS2019 Enterprise - `https://portal.azure.com/`

Cross-platform Startup SPA Solution:

Not so Easy

IDE:

- VS2019 Enterprise - `https://portal.azure.com/`
- Clion with Make

Target Environment: ▶ Windows ▶ MacOS ▶ Linux

C++ Development

All teams **must** use C++17, only STL *without* external libraries.
If teams wish to make any changes, they must seek permission.

- Compiling:
  - Turning source code into object code
  - VS2019: Additional Include Directories
- Linking:
  - Combining all the object code with the libraries into binaries
  - VS2019: Additional Dependencies
- Building:
  - The whole sequence from compiling to linking

Your tutors will be added, your Github group will be monitored.

C++ Development

There are usually 2 build targets:

- Debug:
  - PDB files are created: a lot more code added into your code to enable debugging
  - Not optimized, when compared to Release
- Release:
  - Optimized
  - What we will run our tests on

Note: We will use x86 Debug and Relase and not x86_64

Script check-submission.py

Its a python tool that you should run to make sure your project satisfy the basic submission requirements (you should still check the requirements)!

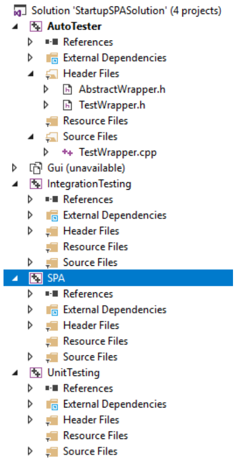
```
$ python check—submission.py
This script will check for basic compliance with the submission requirements.
Disclaimer: you are still responsible for your submission, this check is by
no means complete.
——————————————————————————————————————————————————————————————————————————————
[Failed] — Team number must be valid.
```

spa-win: Visual Studio

Notes:

- All will need to run on Windows with Visual Studio 2019.
- The easy way to start!
- Official and recommended development
- GUI Development is avaliable (optional)
- Easy and integrated test framework - VS Unit Testing
- Members on MacOS/Linux will need to run Windows with VS

spa-win: Visual Studio


Solution 'StartupSPASolution' (4 projects)
AutoTester
References
External Dependencies
Header Files
AbstractWrapper.h
TestWrapper.h
Resource Files
Source Files
TestWrapper.cpp
Gui (unavailable)
IntegrationTesting
References
External Dependencies
Header Files
Resource Files
Source Files
SPA
References
External Dependencies
Header Files
Resource Files
Source Files
UnitTesting
References
External Dependencies
Header Files
Resource Files
Source Files

- **AutoTester**: Build to get AutoTester.exe for your SPA
- **IntegrationTesting**: Implement IntegrationTest.cpp and test classes, build and run for integration testing for your SPA
- **Empty SPA project**: Fill in the code for your SPA
- **UnitTesting**: Implement UnitTest.cpp and test classes, build and run for unit testing for your SPA

Notes: Do not delete the included spa libraries that come with the repository.

spa-win: Visual Studio

```
> AutoTester.exe ..\Tests\Sample_source.txt ..\Tests\Sample_queries.txt out.xml
```

1. Running autotester:
   1.1 Build in VS then, from the command line (cmd in Windows)
   1.2 From VS in Debug mode, see run settings for parameters
2. Open out.xml in Mozilla Firefox to see the results

Note:

- analysis.xsl is the stylesheet and should be in the same directory with out.xml
- Depending on your Firefox version, you will need to go `about:config` in the URI path and change the property `security.fileuri.strict_origin_policy` to False

spa-cp: CMake

We strongly recommend you to use the same OS/IDE/Build system across every member! Make sure your team all agree on and declare a target platform from the following:

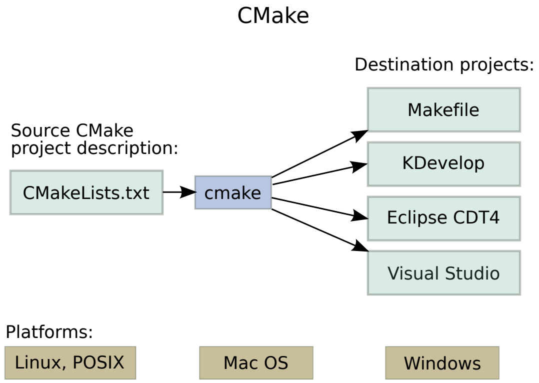
1. Windows / VS / CMake
2. MacOS / AppleClang / CMake
3. Linux (Fedora) / GCC / CMake

Notes:

- You can mix OS/IDE/Build Systems; no additional marks will be given. Support will be given on best-effort basis.
- GUI is not recommended and not supported on CMake. Support will be given on best-effort basis.
- Catch is the testing framework used.

spa-cp: CMake

Make sure you check out the tutorial: `https://cmake.org/cmake/help/latest/guide/tutorial/index.html`[1]



[1]Img from `https://www.shlomifish.org/lecture/CMake`

spa-cp: CMake

Similar project structure as spa-win.

- **AutoTester**: Build to get AutoTester.exe for your SPA
- **IntegrationTesting**: Implement IntegrationTest.cpp and test classes, build and run for integration testing for your SPA
- **Empty SPA project**: Fill in the code for your SPA
- **UnitTesting**: Implement UnitTest.cpp and test classes, build and run for unit testing for your SPA

Notes: Do not delete the included spa libraries that come with the repository.

spa-cp: CMake

1. Build and Compile using your platform instructions.
2. Running from the command line (cmd in *nix)

```
> ./autotester ../../../tests/Sample_source.txt
  ../../../tests/Sample_queries.txt
  ../../../tests/out.xml
```

Note:

- analysis.xsl is the stylesheet and should be in the same directory with out.xml
- Depending on your Firefox version, you will need to go `about:config` in the URI path and change the property `security.fileuri.strict_origin_policy` to False

Q & A

Some parting advice:

- Use spa-win for quick tools setup; spa-cp if you want a challenge.
- Always communicate with the teaching team and your teammates; bring up problems early.
- Remember that this module is double the credits and its a difficult.
- If you can learn to work together in a team, this module will be very fulfilling.

The team leader of each team *must* register at `https://github.com/nus-cs3203/project-wiki/wiki/Version-Control-System-and-Code-Repository`.