

Acceptance Test-Driven Development with **Robot Framework**



Agenda



Day 1

Acceptance **T**est-**D**riven **D**evelopment

Concept of Robot Framework

Installation

Architecture of Robot Framework



Day 1

Create test case

Execute test case

How to write **better** test case ?



Day 2

Data Driven Testing

Test template

Working with command line

APIs testing

Integration with **C**ontinuous **I**ntegration

Custom test report



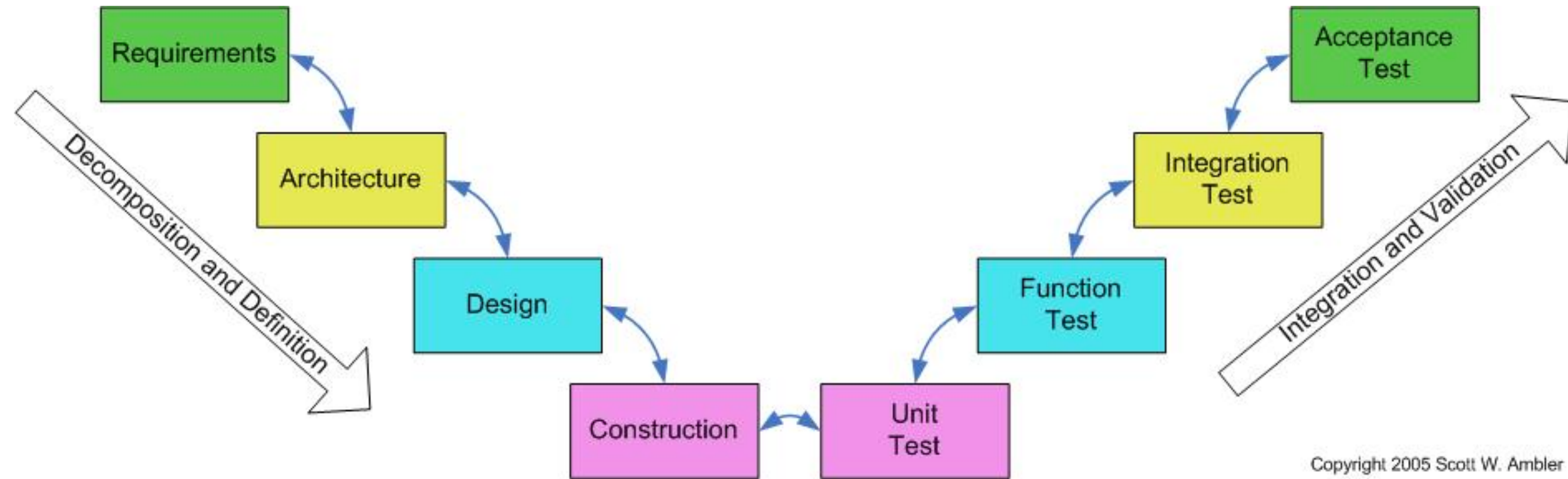
Let's start



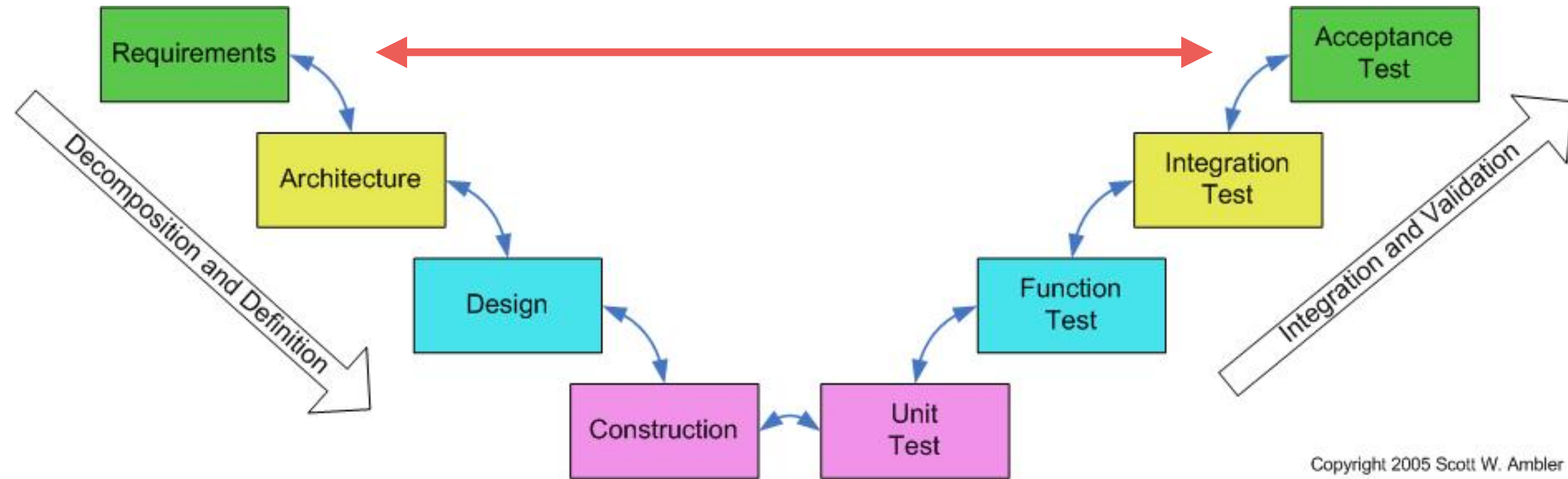
Write your Goal(s) !!



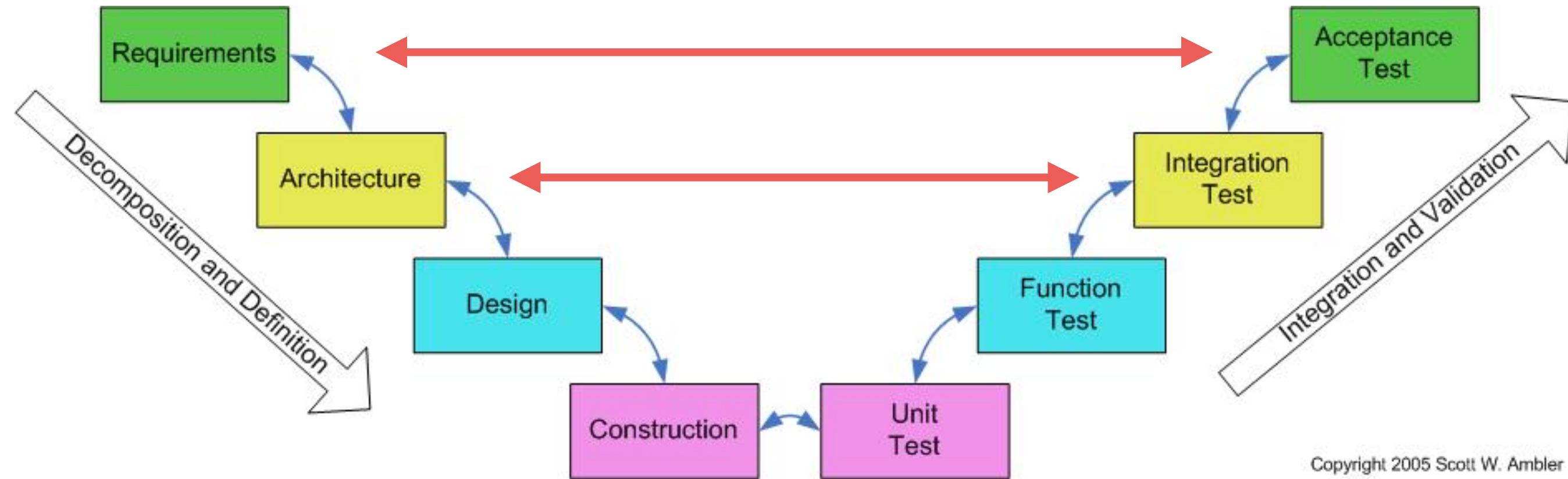
V-Model



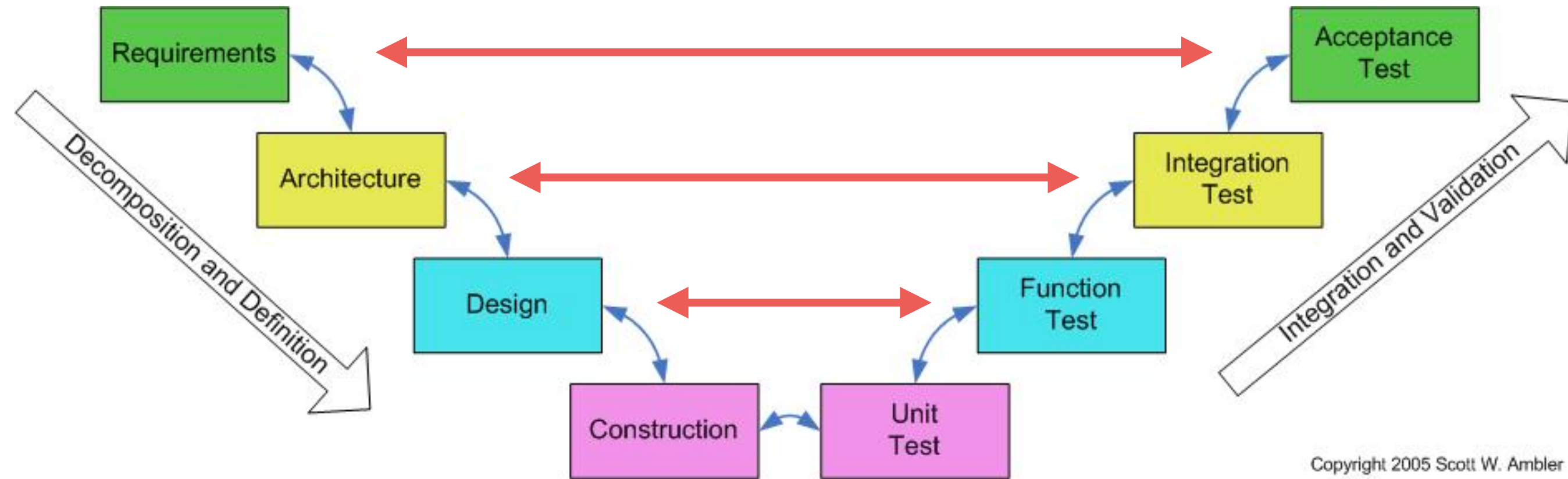
V-Model



V-Model



V-Model



Acceptance Test-Driven Development



ATDD

Common understanding

Common language

Executable requirements or examples

Living documentation



Same goal !!

Story **D**riven **D**evelopment

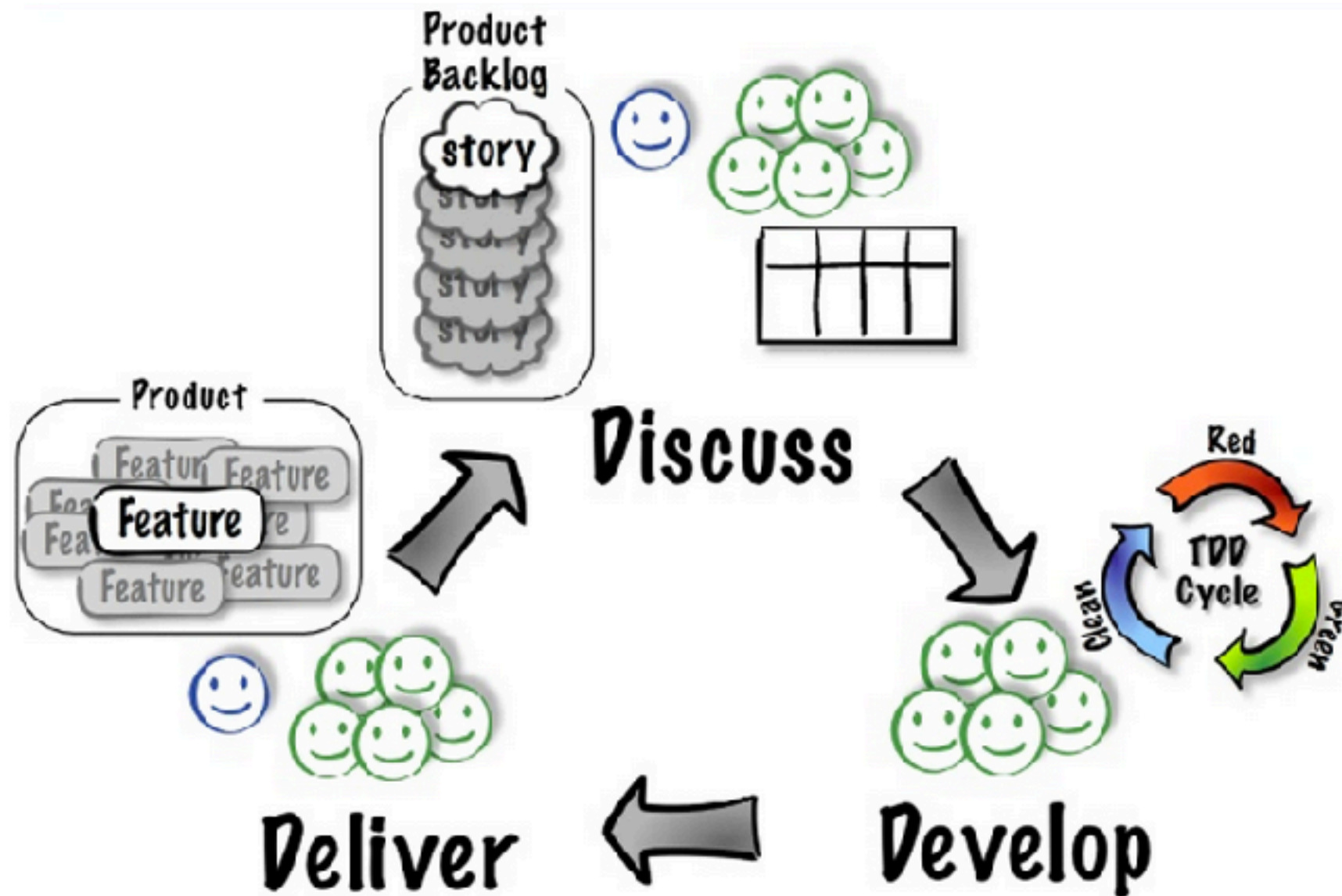
Behavior **D**riven **D**evelopment

Specification **b**y **E**xample

Example **D**riven **D**evelopment



ATDD cycle

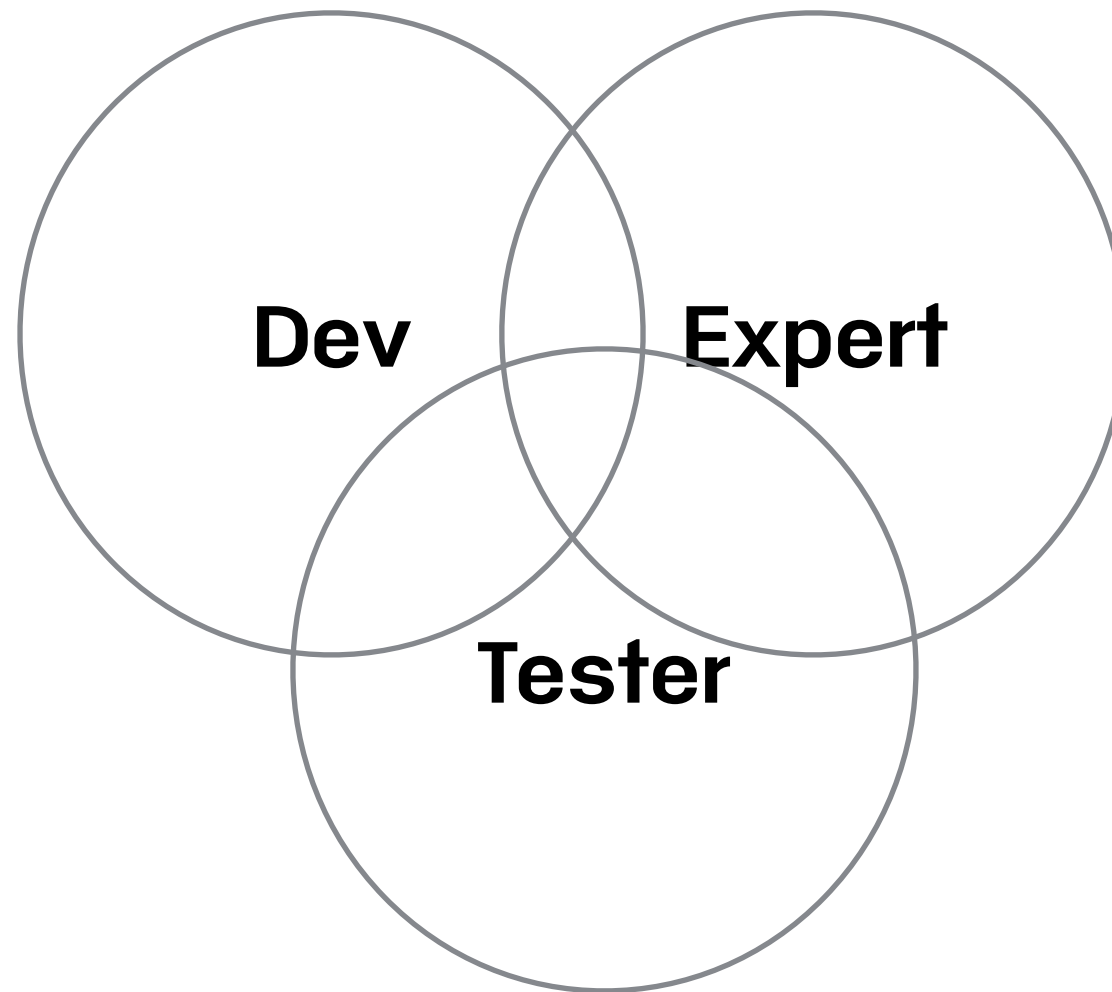


(Model developed with Pekka Klärck, Bas Vodde, and Craig Larman.)



1. Discuss

Whole team approach



1. Discuss

Whole team approach

Sharing understanding

Clarify the solution

Concrete examples

Examples can be use in **automated !!**



2. Develop

Implement work with specified examples

Automation of the examples

Whole team is responsible on automation

At the end, all automated examples **pass**



3. Deliver

Features are **demonstrated** to all stakeholders

All examples pass (new + existing)

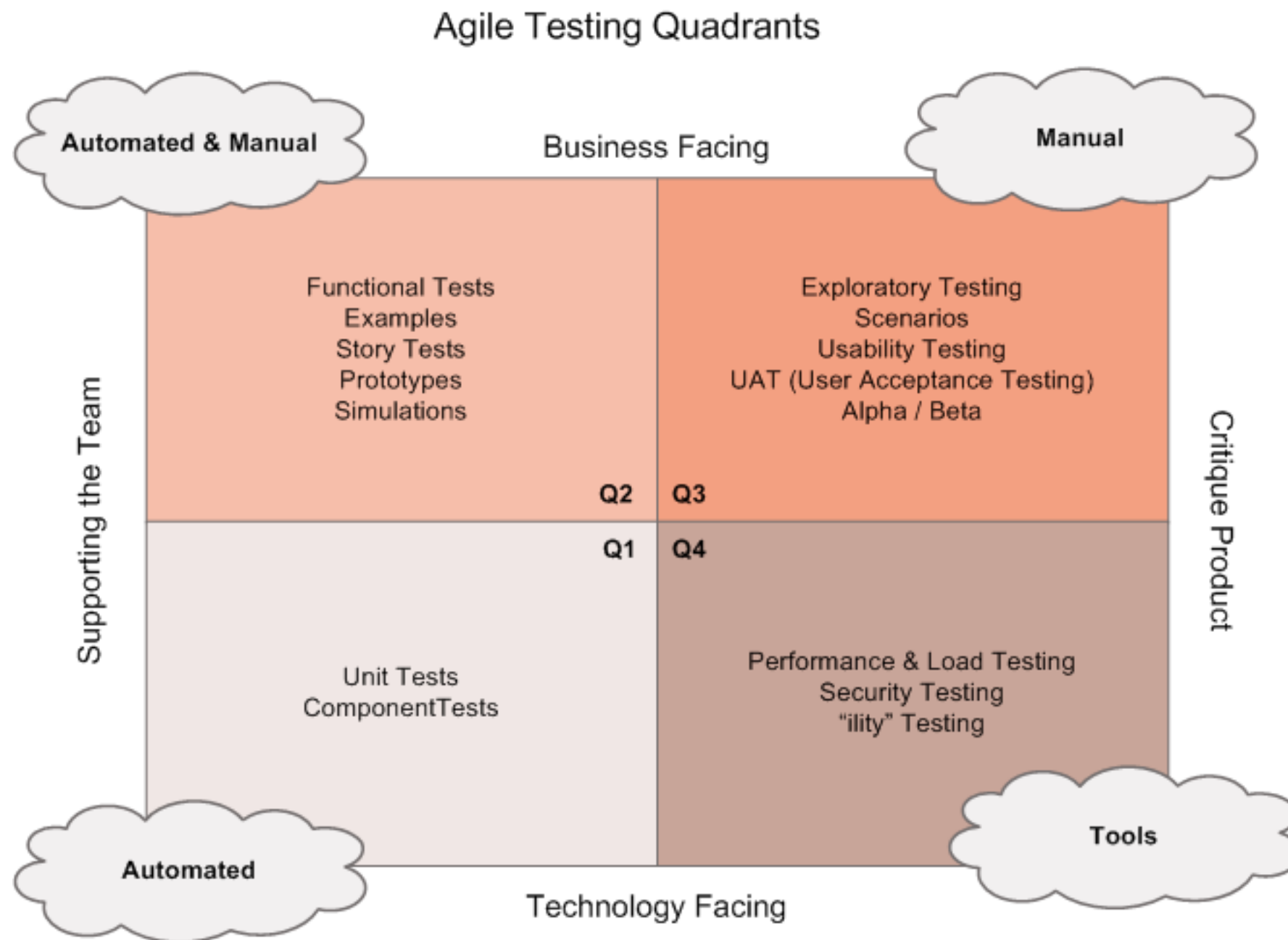
Feedback as input to the next discuss



Acceptance Tests
=
Acceptance Criteria
+
Examples(data + scenarios)



Agile Testing



Robot Framework



Introduction

Test automation framework

Designed for Acceptance testing(ATDD)

Developed with Python

Compatible with Java (Jython)



History

Started from **Pekka Klärck's** masters thesis

HELSINKI UNIVERSITY
OF TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author: Pekka Laukkanen

Name of the thesis: Data-Driven and Keyword-Driven Test Automation Frameworks

Date: February 24, 2006

Number of pages: 98 + 0

Department: Department of Computer
Science and Engineering

Professorship: T-76

Supervisor: Prof. Reijo Sulonen

Instructor: Harri Töhönen, M.Sc.



History

Initialed in 2005 by Nokia Siemens Network

Opensource in 2008

Created robotframework.org

Big communities at [Github](https://github.com)



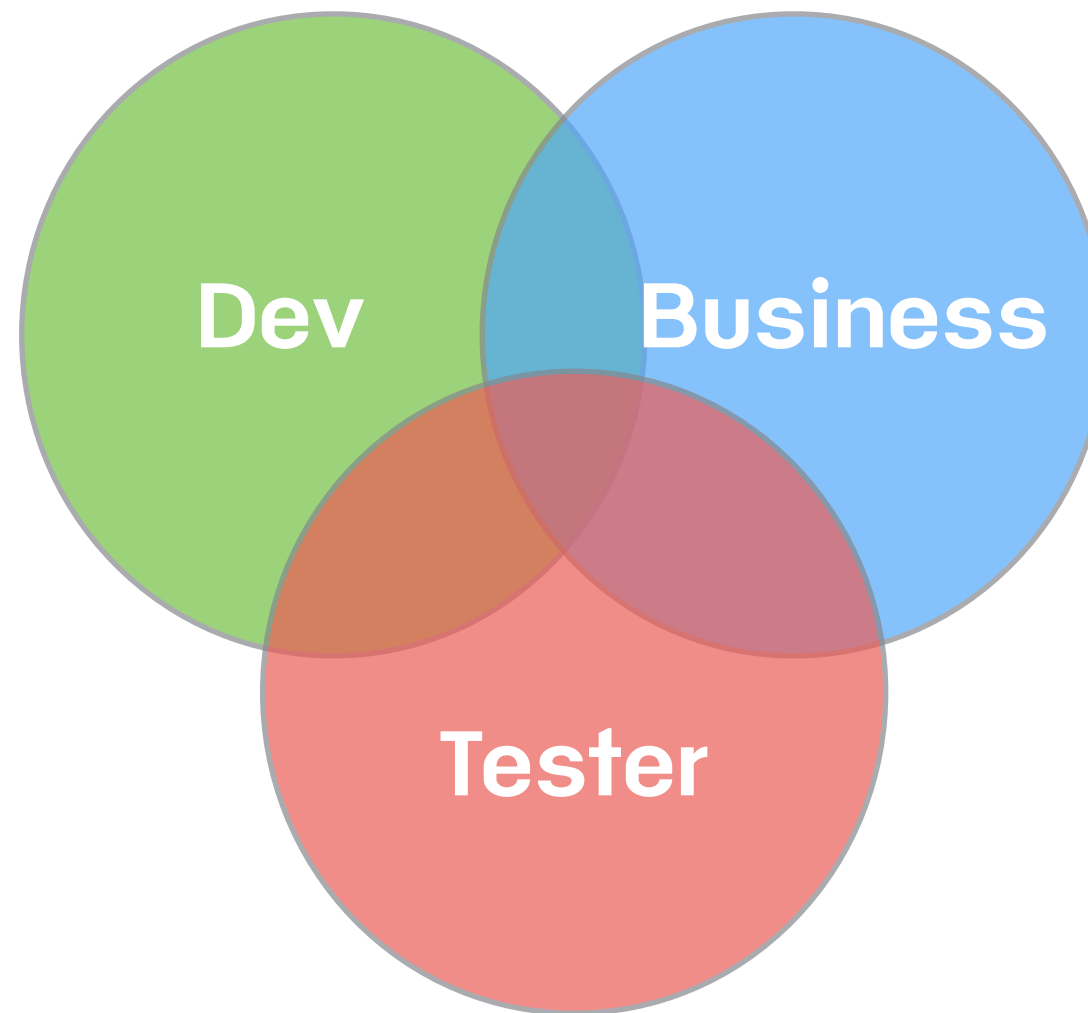
Concepts

Acceptance Testing

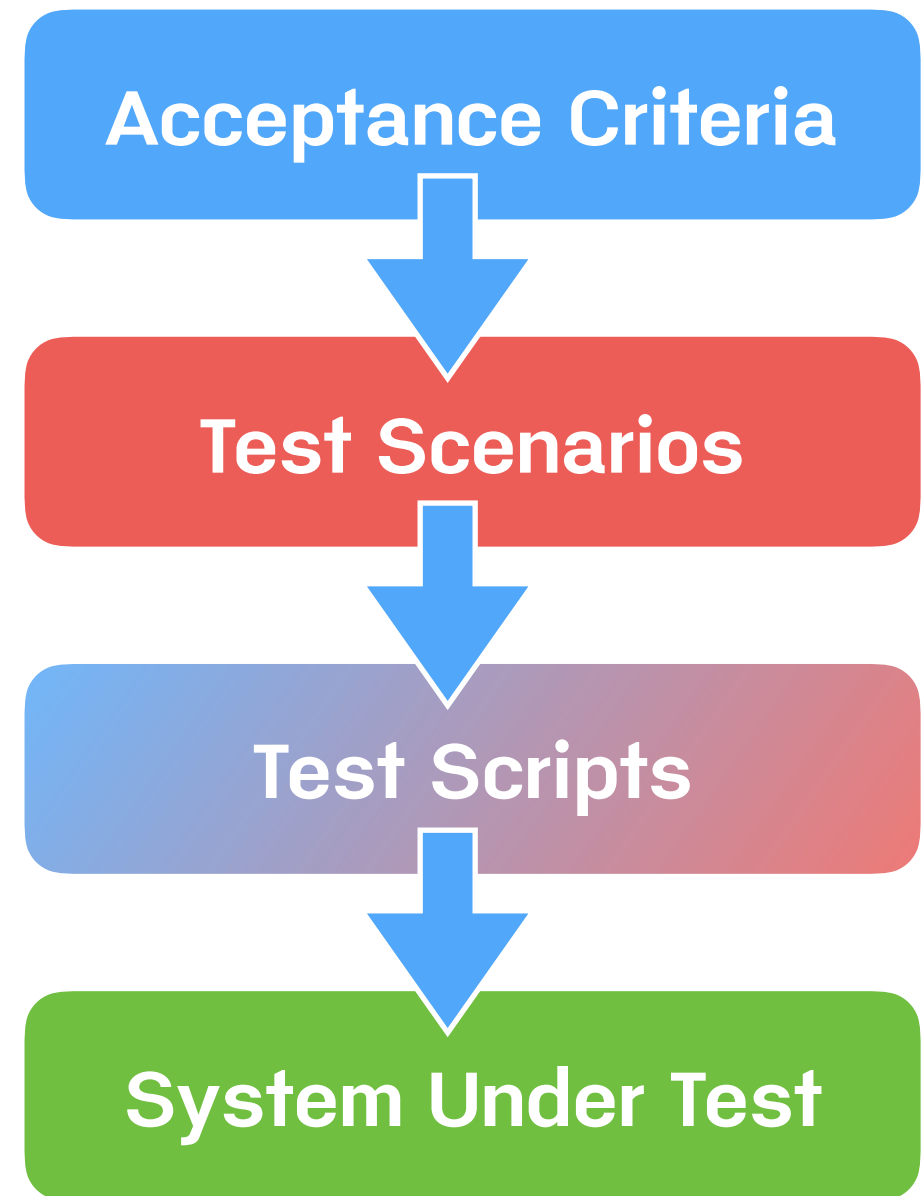
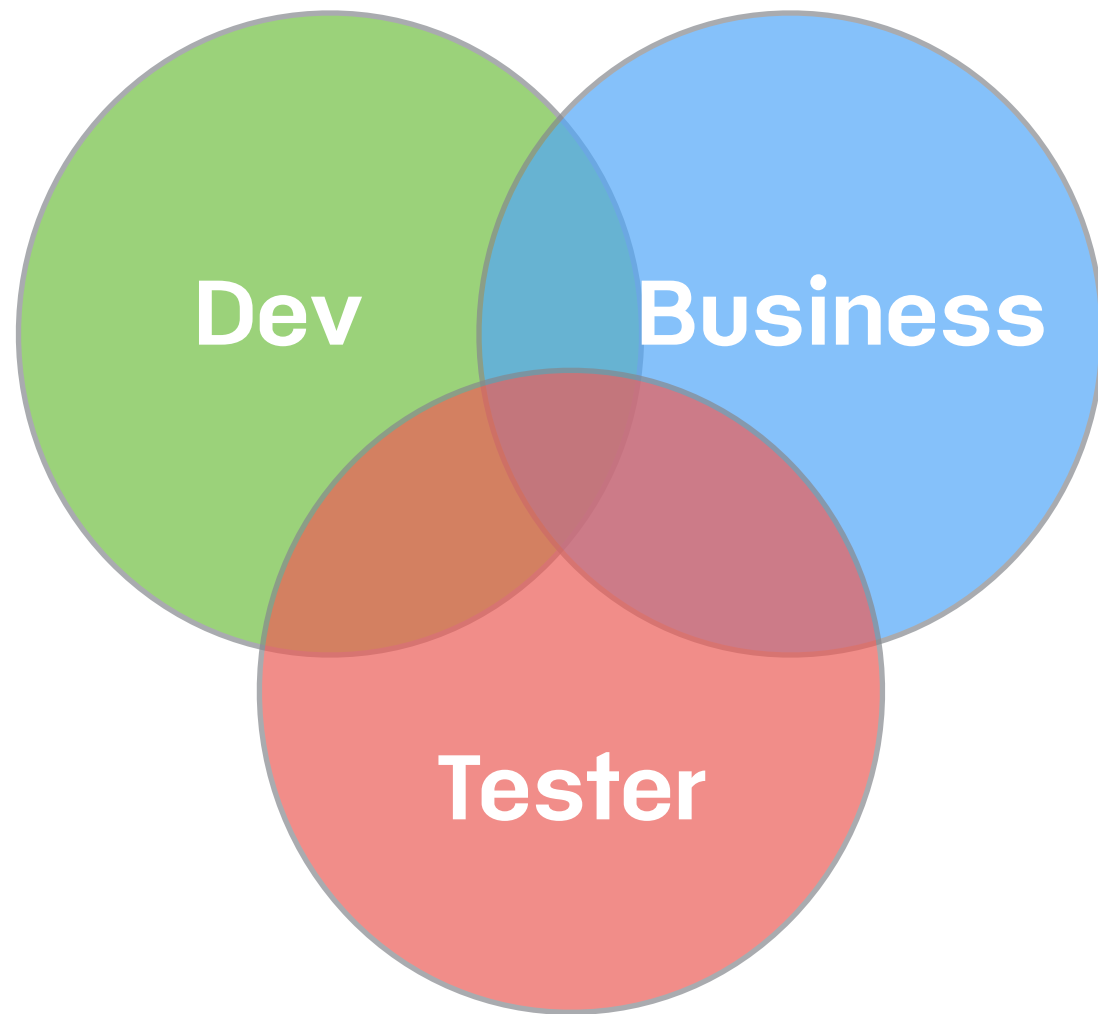
Keyword Driven Development



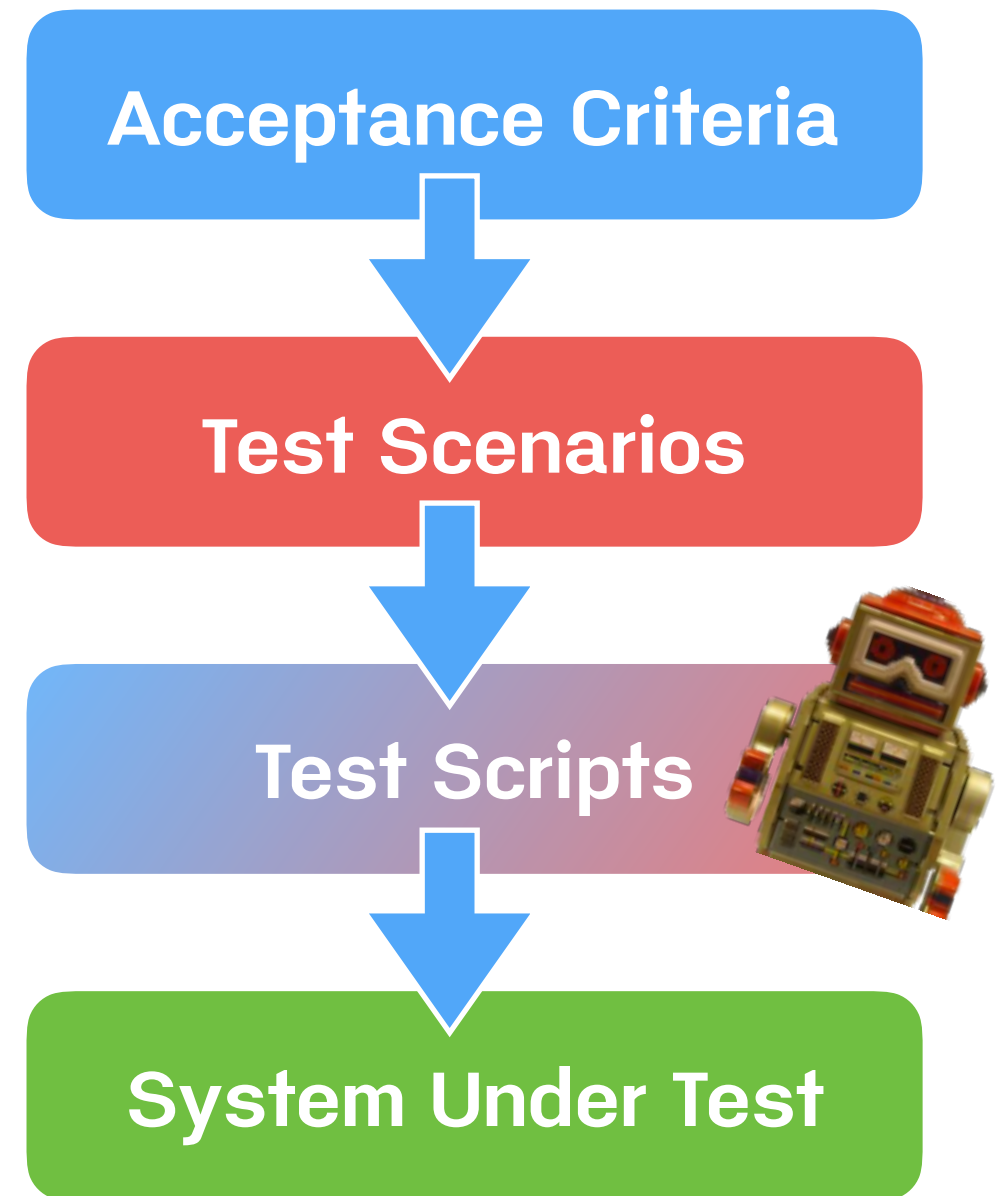
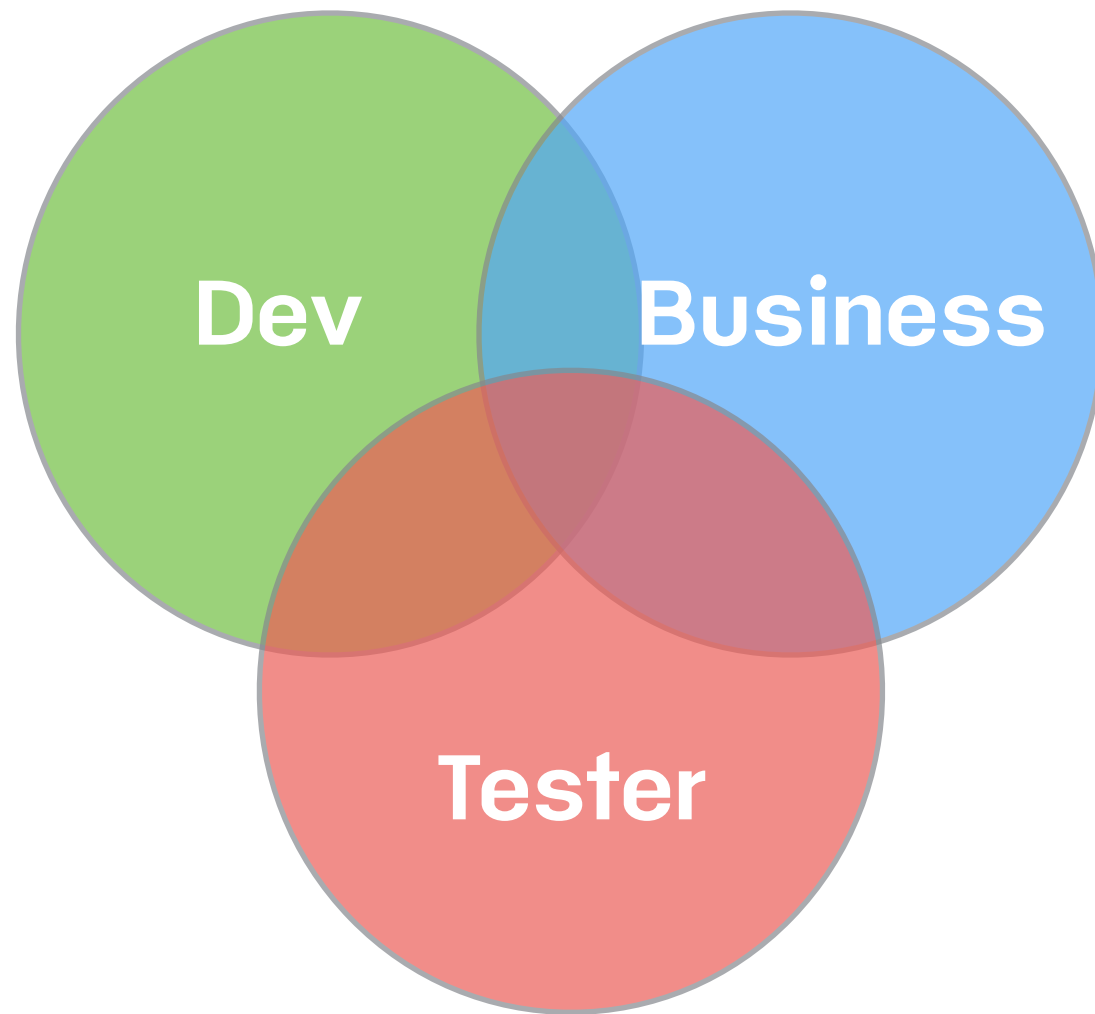
Acceptance Testing



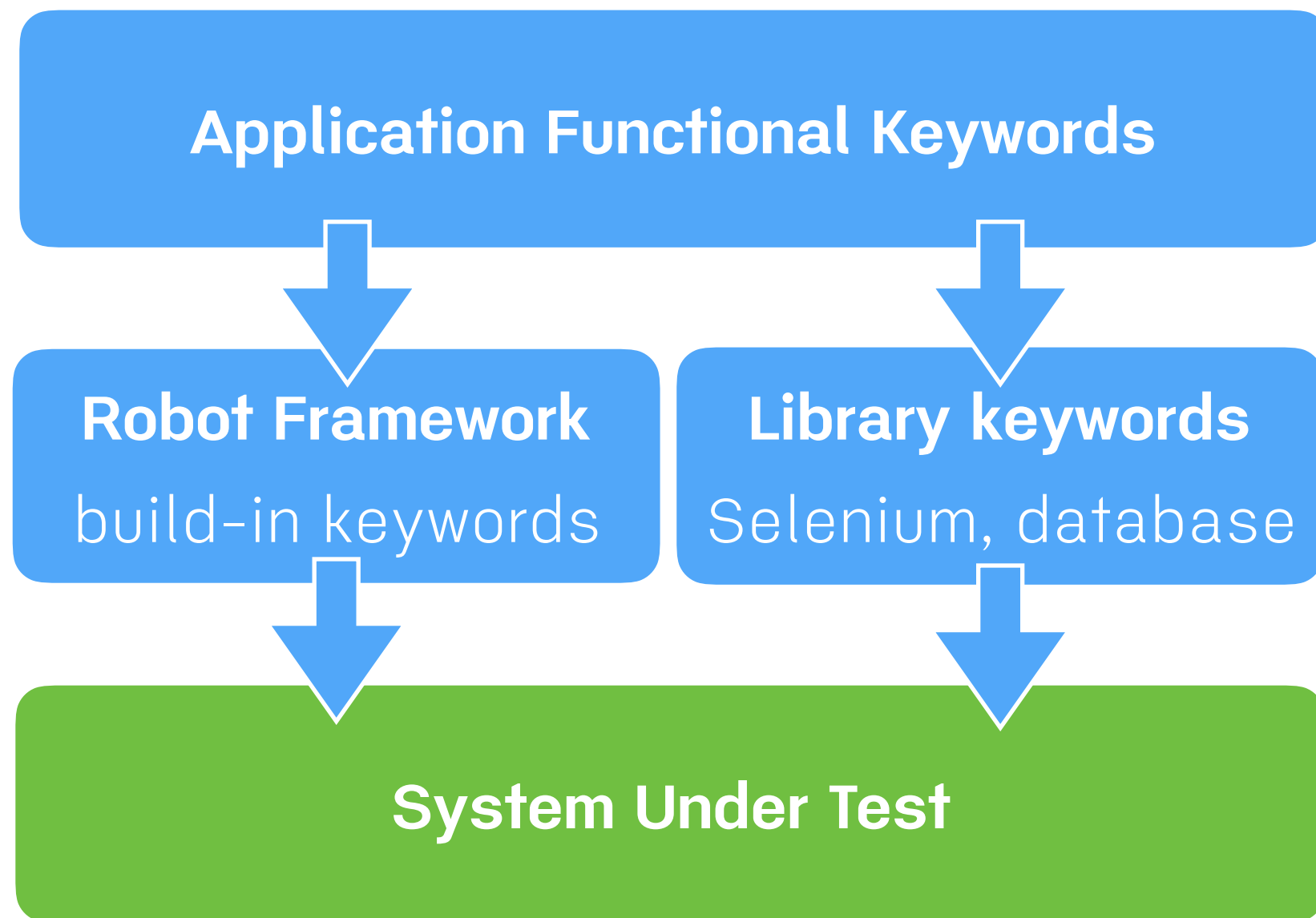
Acceptance Testing



Acceptance Testing



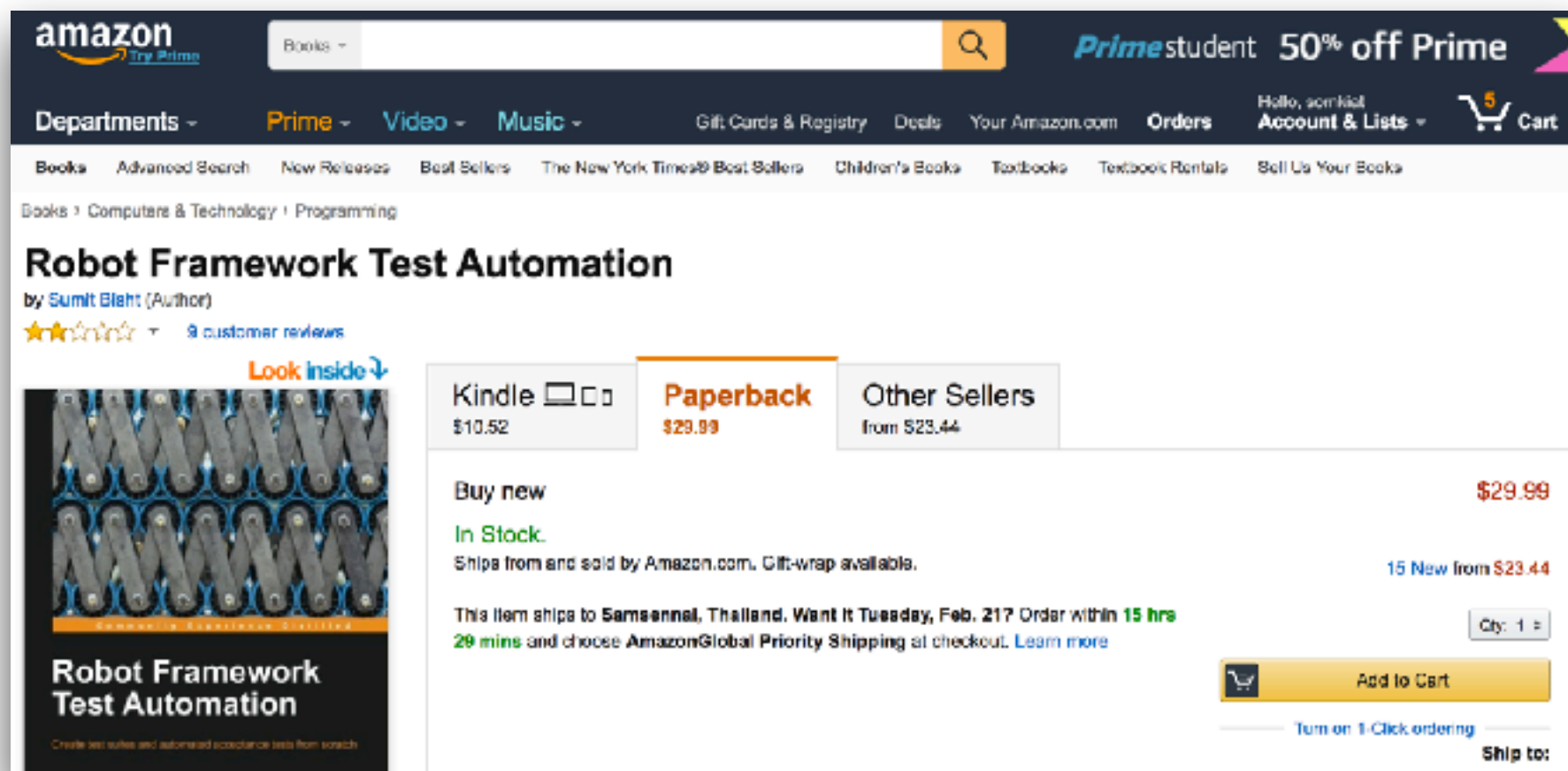
Keyword Driven Development



Keyword Driven Development

Example scenario

“Add product to cart”



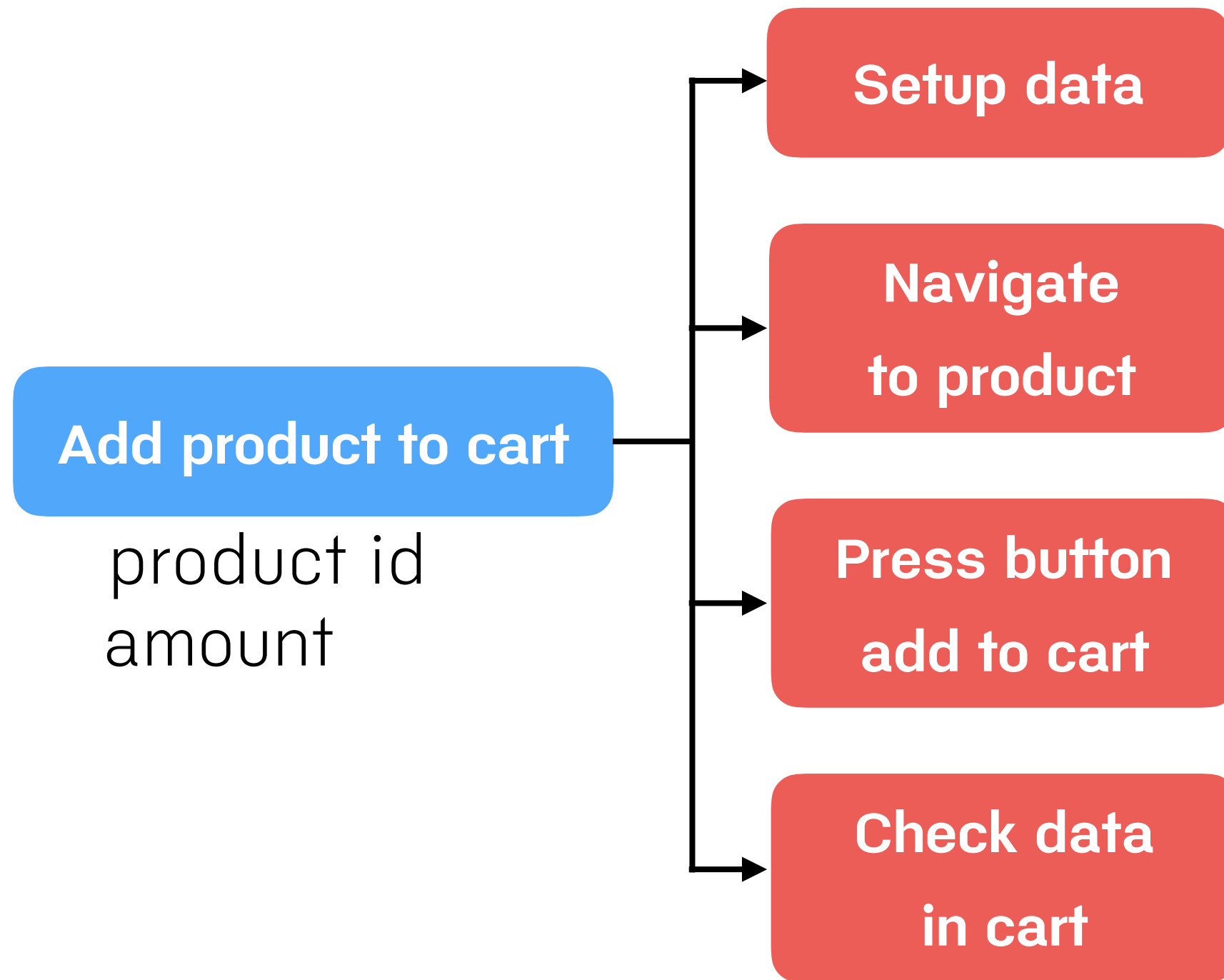
Add product to cart

Add product to cart

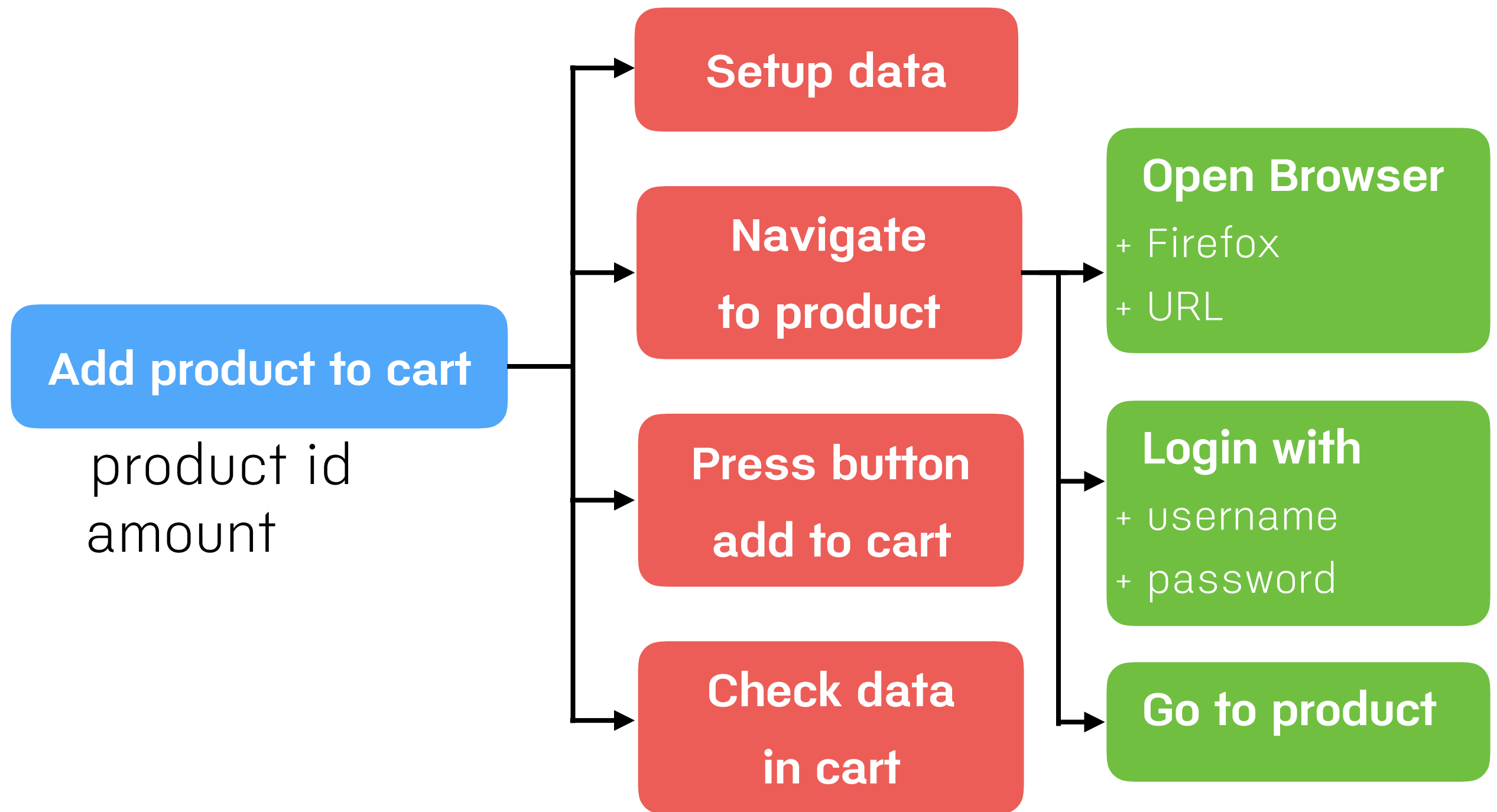
product id
amount



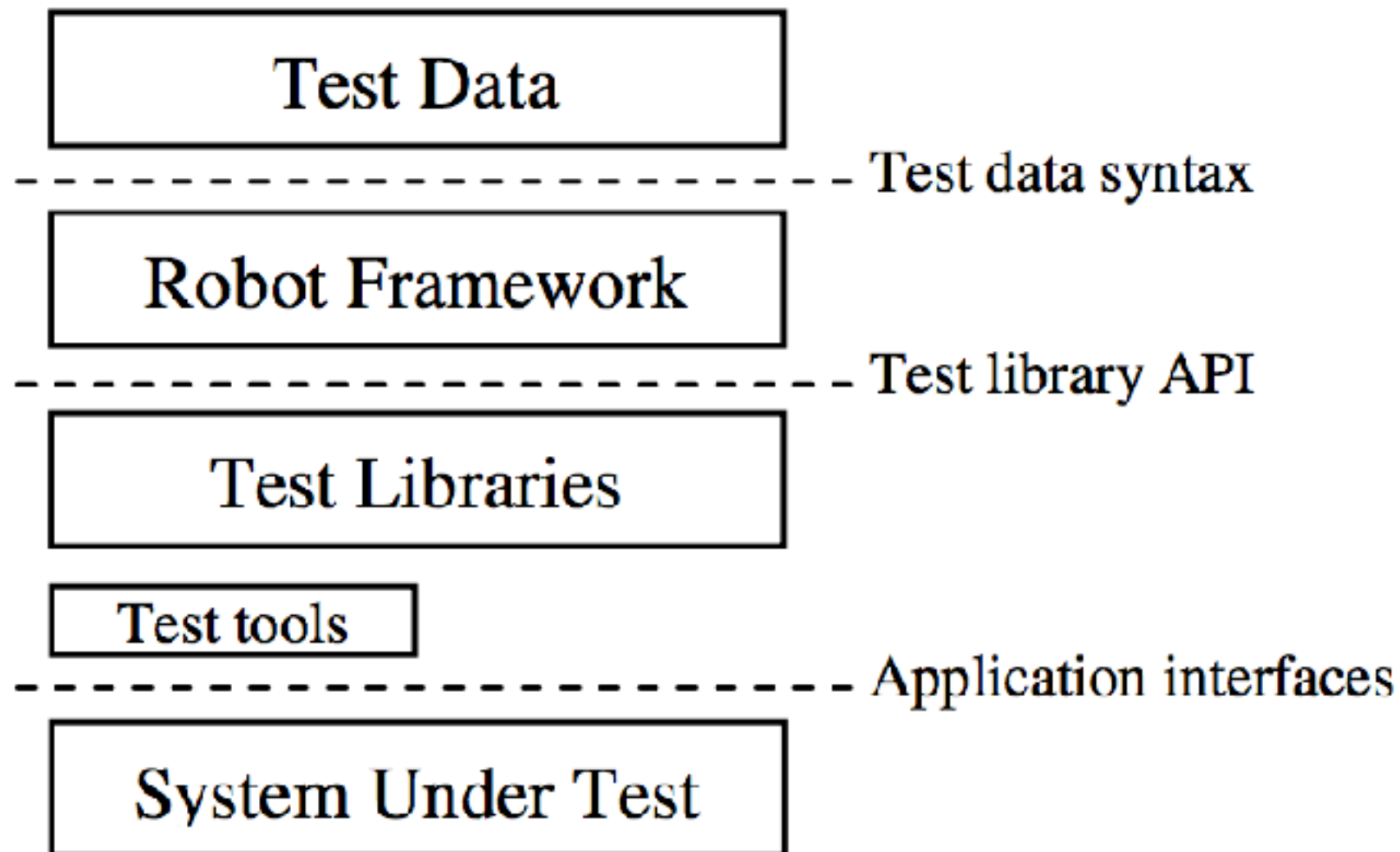
Add product to cart



Add product to cart



High level architecture



Robot Framework Development



Installation

1. Python 2.7.13
2. PIP (Package manager for python)
3. Robot Framework
4. Robot Framework Library
5. IDE and Editor



Install python and pip

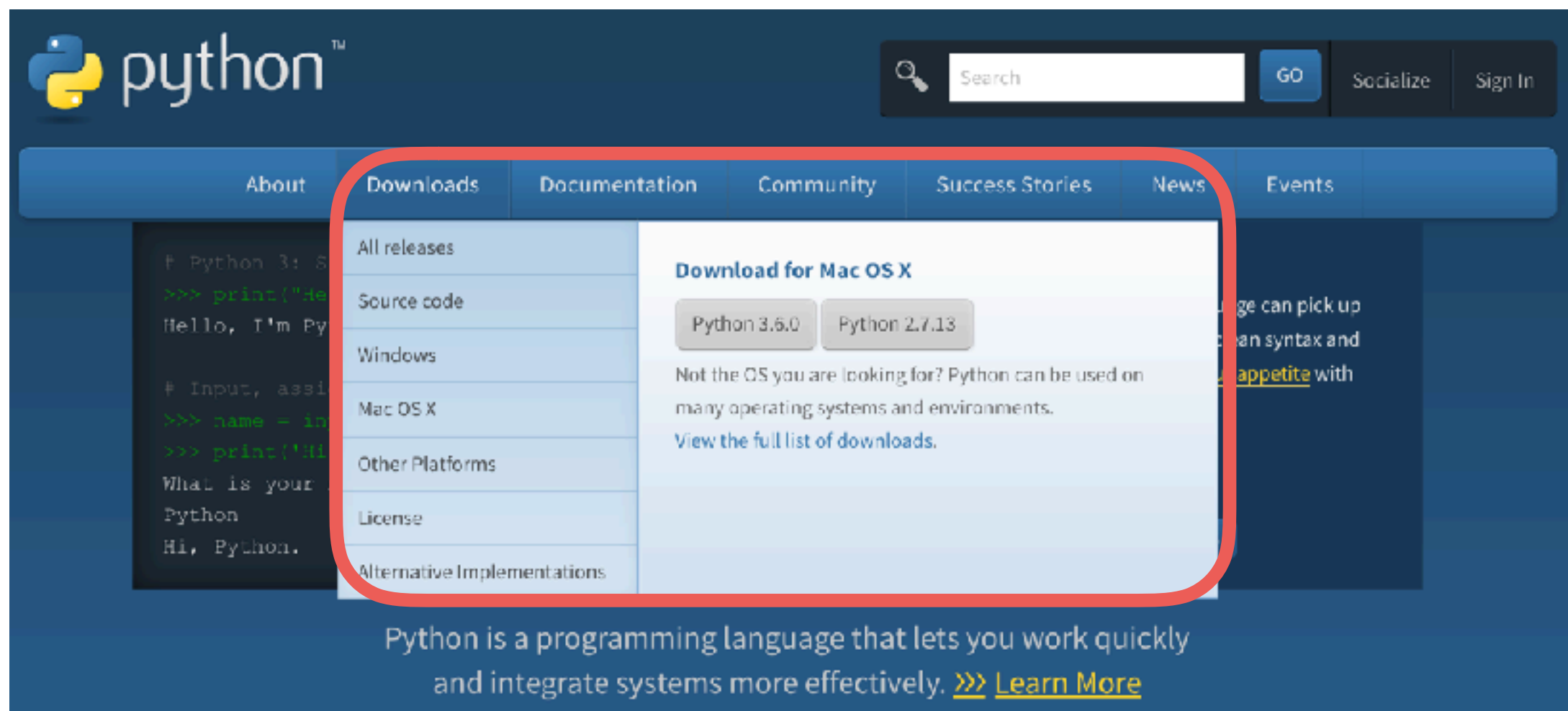


For Windows

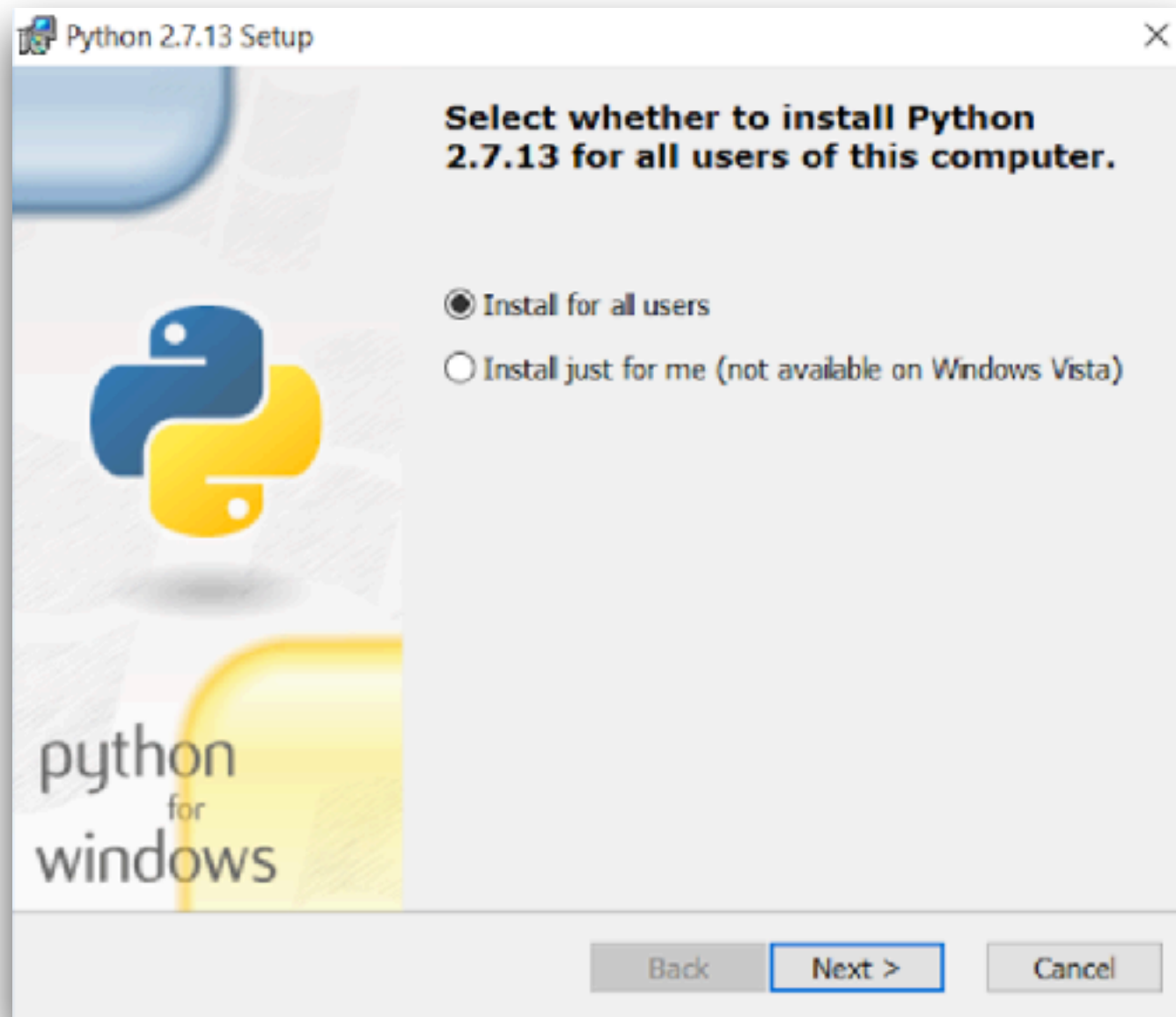


Download Python 2.7.13

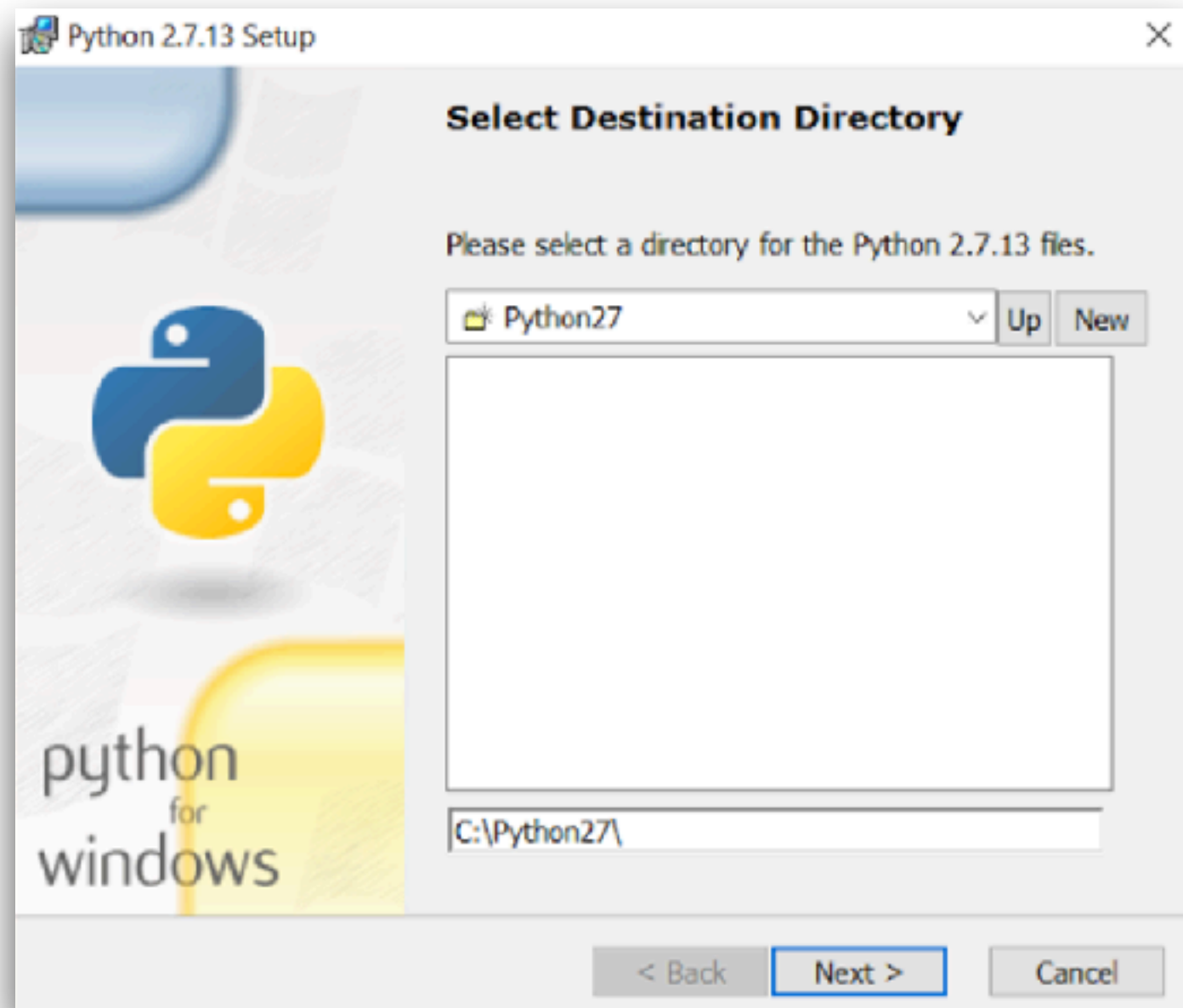
<https://www.python.org/>



Install Python 2.7.13



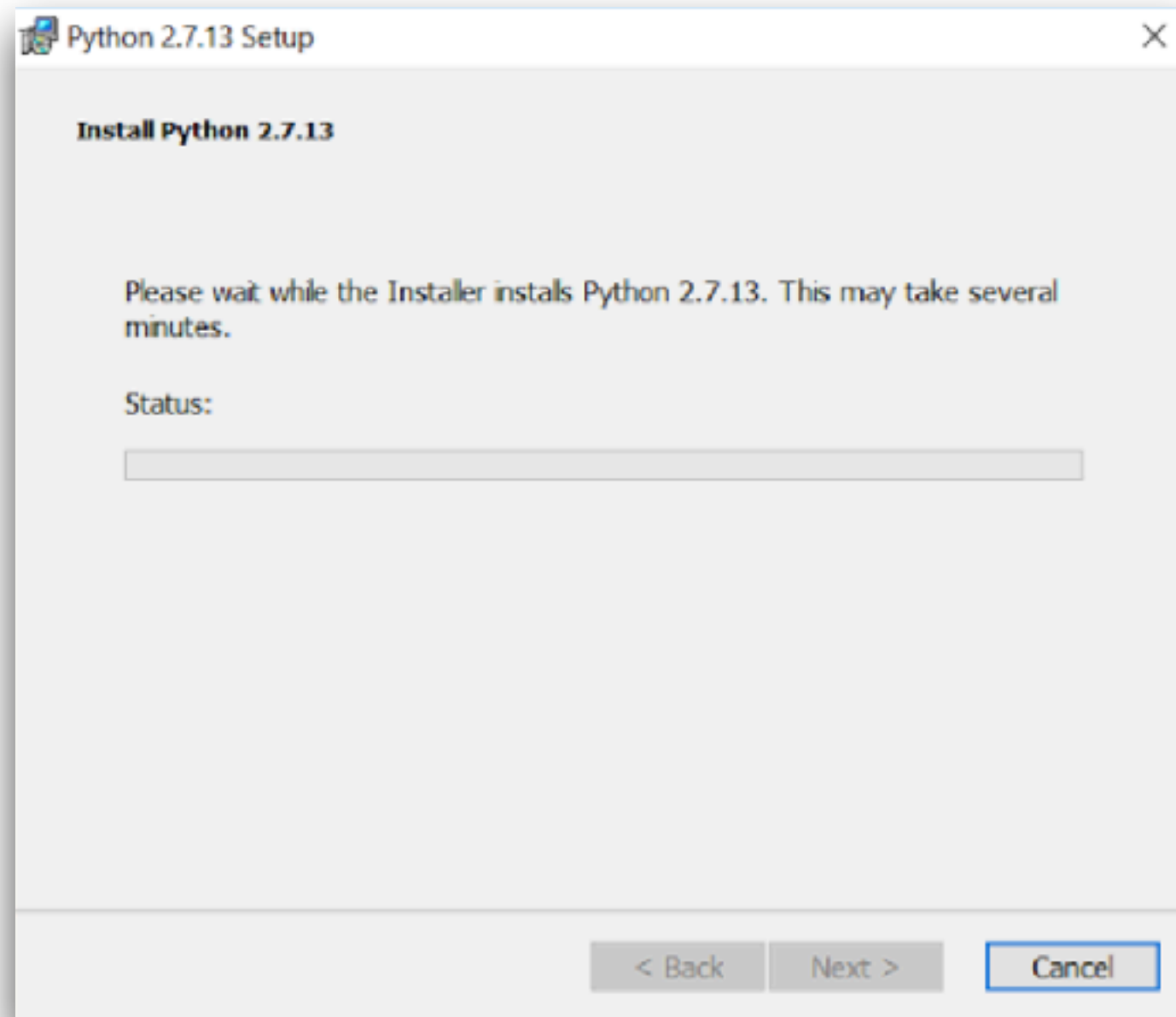
Install Python 2.7.13



Install Python 2.7.13



Install Python 2.7.13



Install Python 2.7.13



Run python in command line

\$python

Command Prompt

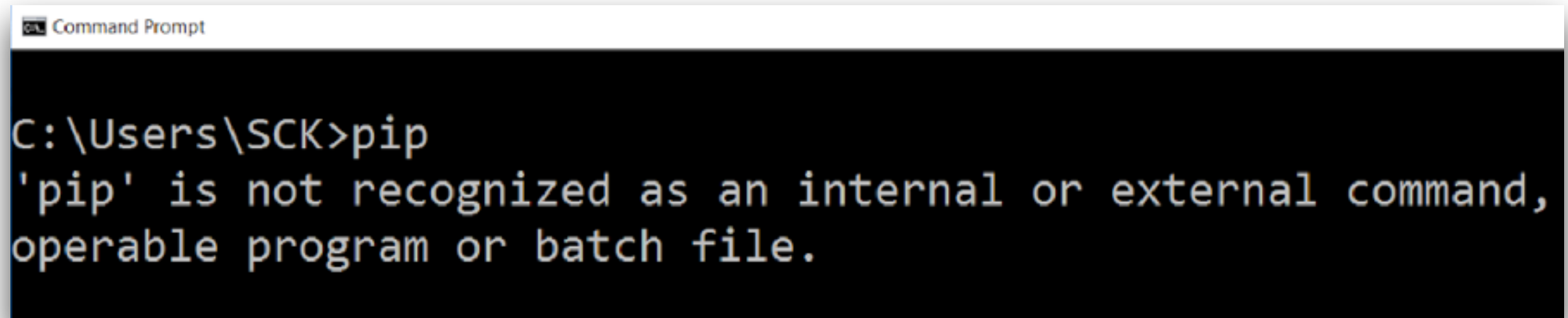
```
C:\Users\SCK>python
```

```
'python' is not recognized as an internal or external command,  
operable program or batch file.
```



Run pip in command line

\$pip



```
Command Prompt
C:\Users\SCK>pip
'pip' is not recognized as an internal or external command,
operable program or batch file.
```



Configuration in command line

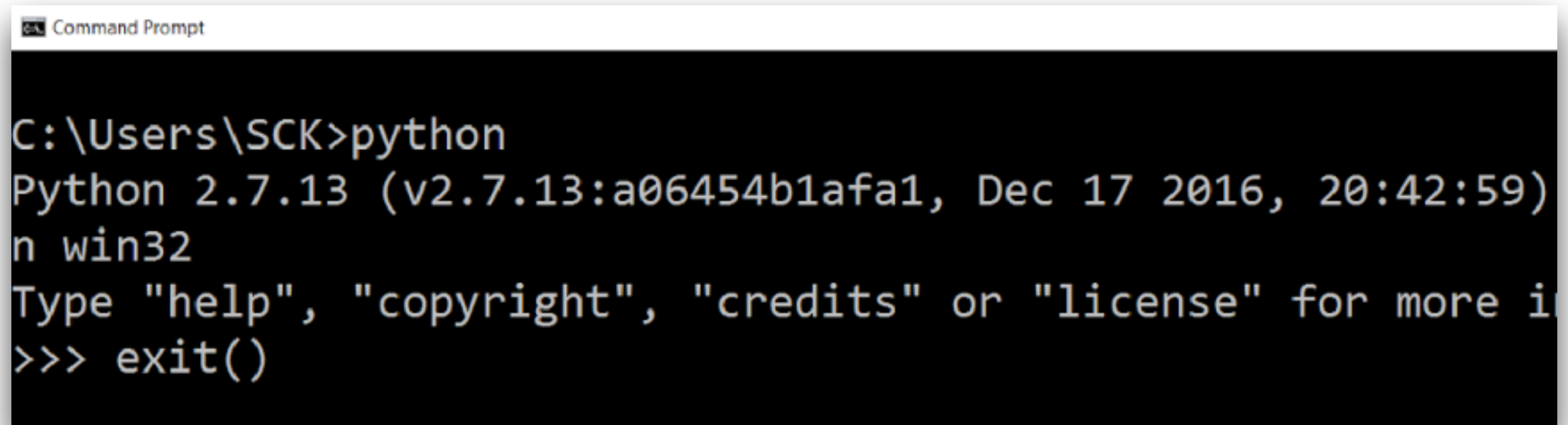
```
$set PYTHON_HOME=c:\python27
```

```
$set PATH=./;%PYTHON_HOME%;%PYTHON_HOME%\Scripts;%PATH%
```



Run python in command line

\$python

A screenshot of a Windows Command Prompt window. The title bar at the top says "Command Prompt". The command prompt shows the following text: "C:\Users\SCK>python", "Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59)", "on win32", "Type \"help\", \"copyright\", \"credits\" or \"license\" for more information", and ">>> exit()".

```
Command Prompt
C:\Users\SCK>python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59)
on win32
Type "help", "copyright", "credits" or "license" for more information
>>> exit()
```



Run pip in command line

\$pip

```
Command Prompt

C:\Users\SCK>pip

Usage:
  pip <command> [options]

Commands:
  install          Install packages.
  download         Download packages.
  uninstall        Uninstall packages.
  freeze           Output installed packages.
  list             List installed packages.
  show            Show information about installed packages.
  check           Verify installed packages.
  search          Search PyPI for packages.
```



Permanent configuration



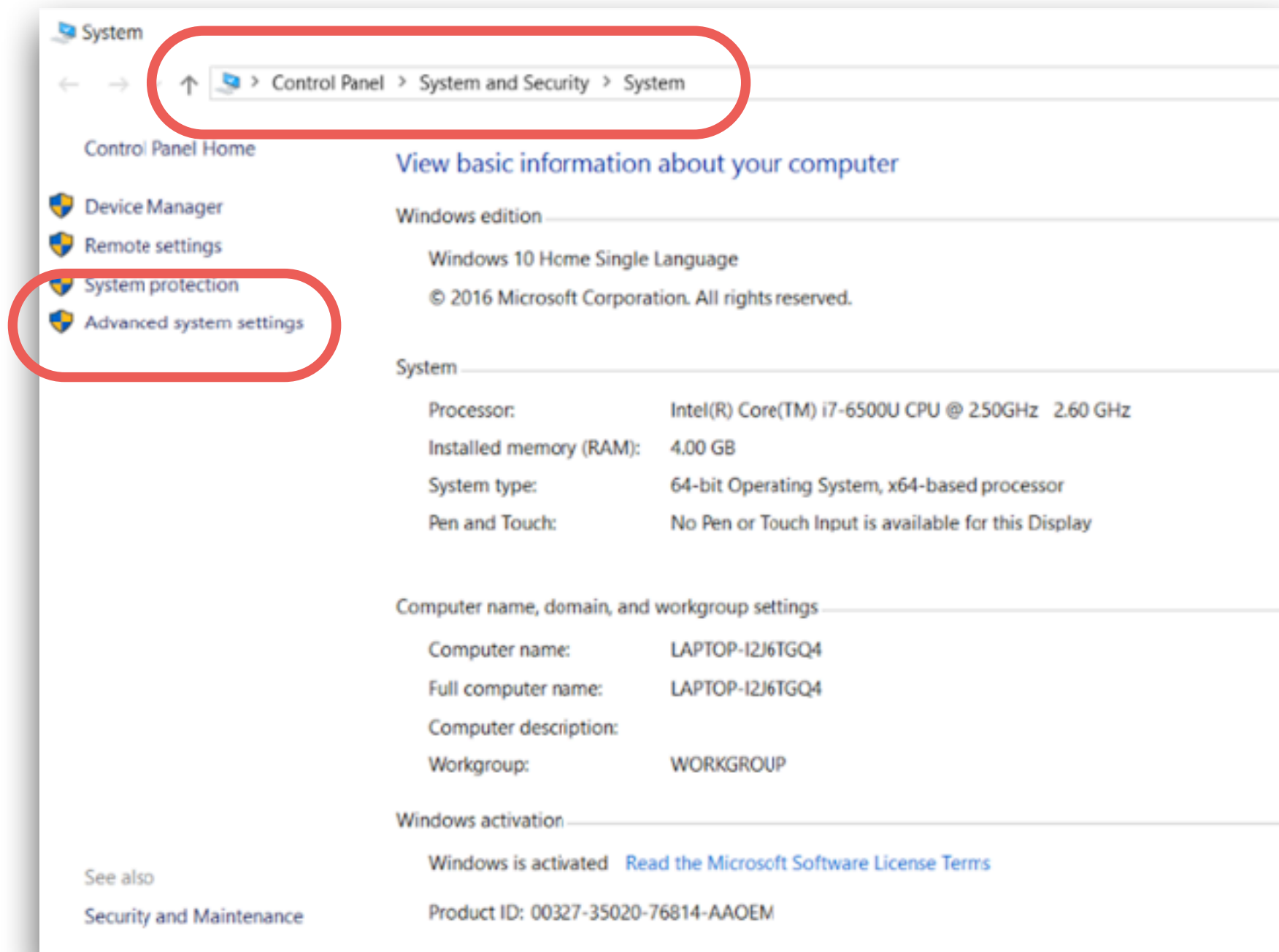
Configuration

Go to Control Panel -> System and Security -> System

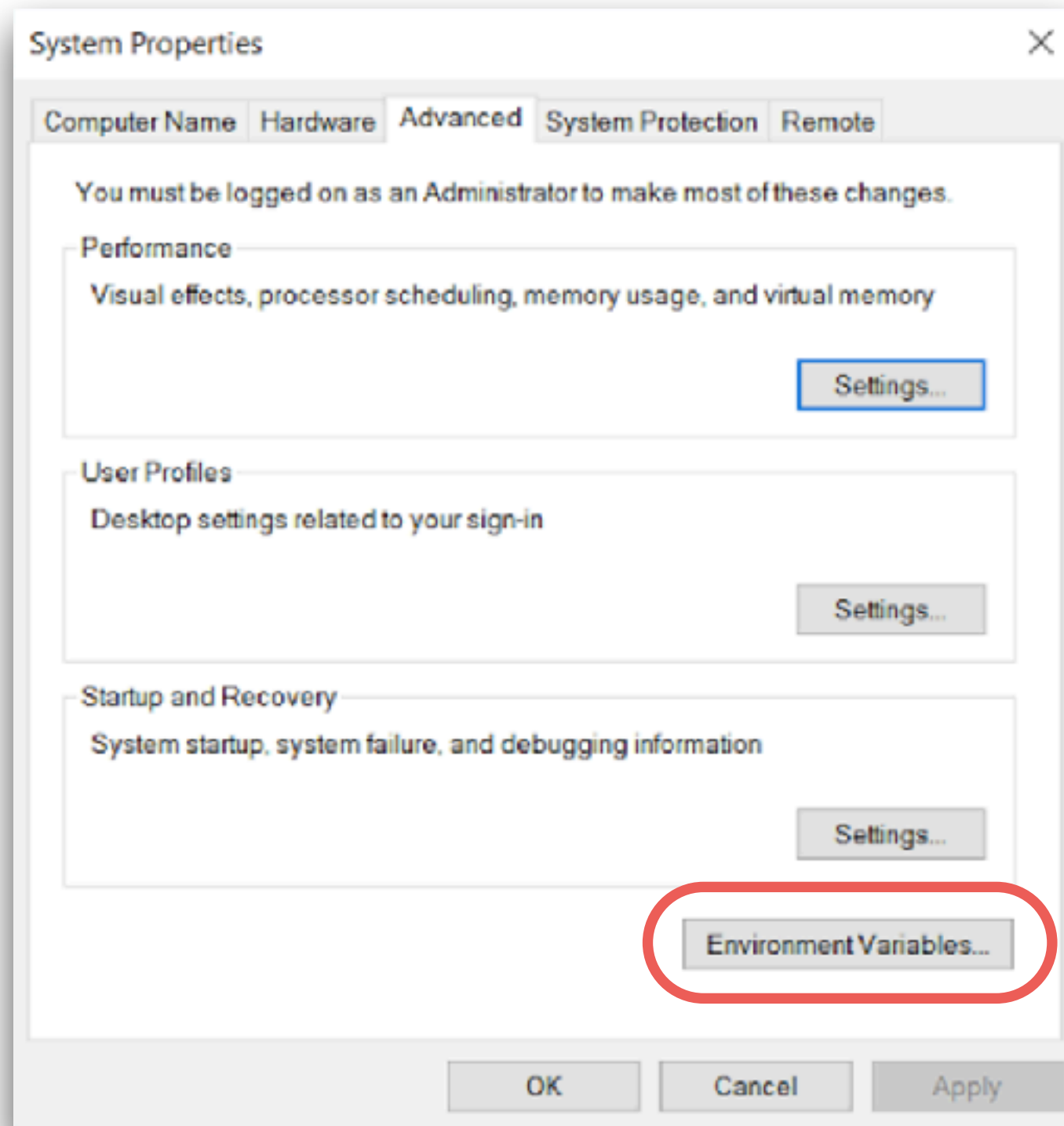
Go to Advanced system settings



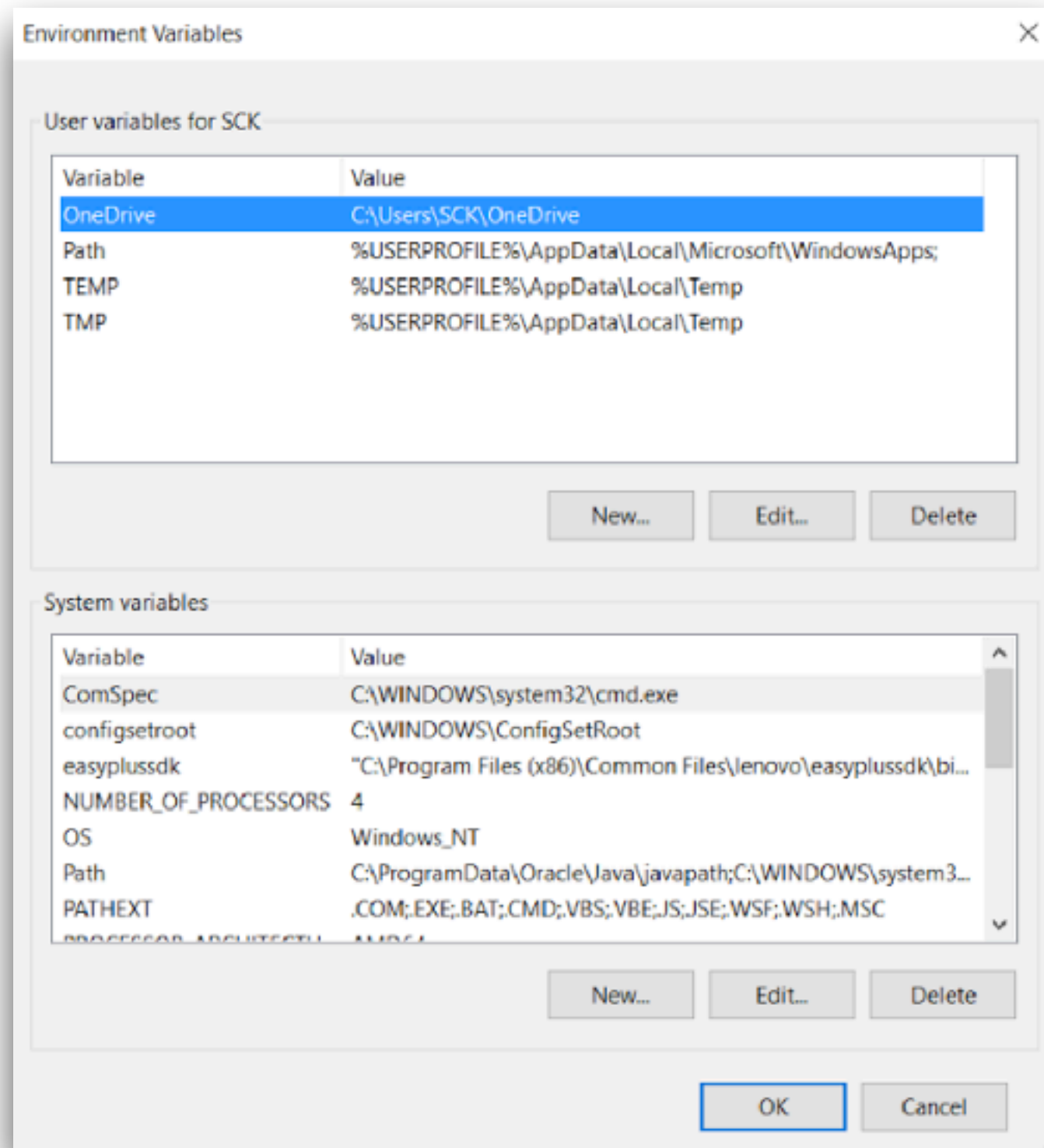
Configuration



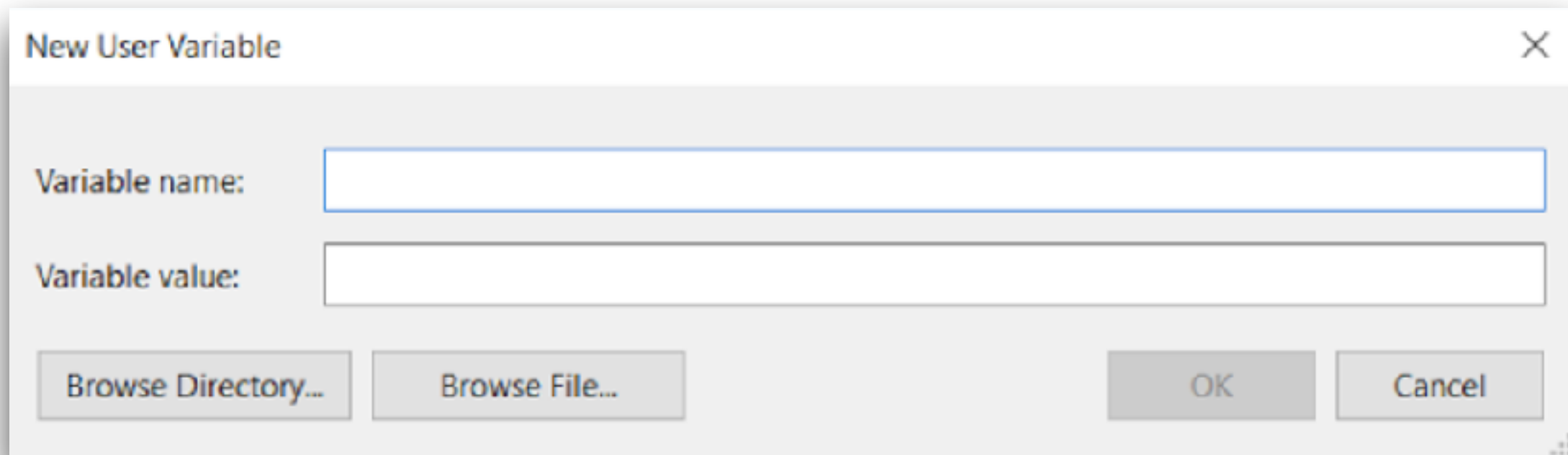
Configuration



Configuration



Configuration



A screenshot of a 'New User Variable' dialog box. The dialog has a title bar with a close button (X) in the top right corner. Inside, there are two text input fields. The first is labeled 'Variable name:' and the second is labeled 'Variable value:'. Below these fields are four buttons: 'Browse Directory...', 'Browse File...', 'OK', and 'Cancel'. The 'Browse Directory...' and 'Browse File...' buttons are on the left, and 'OK' and 'Cancel' are on the right. The dialog box has a light gray background and a subtle shadow.

New User Variable

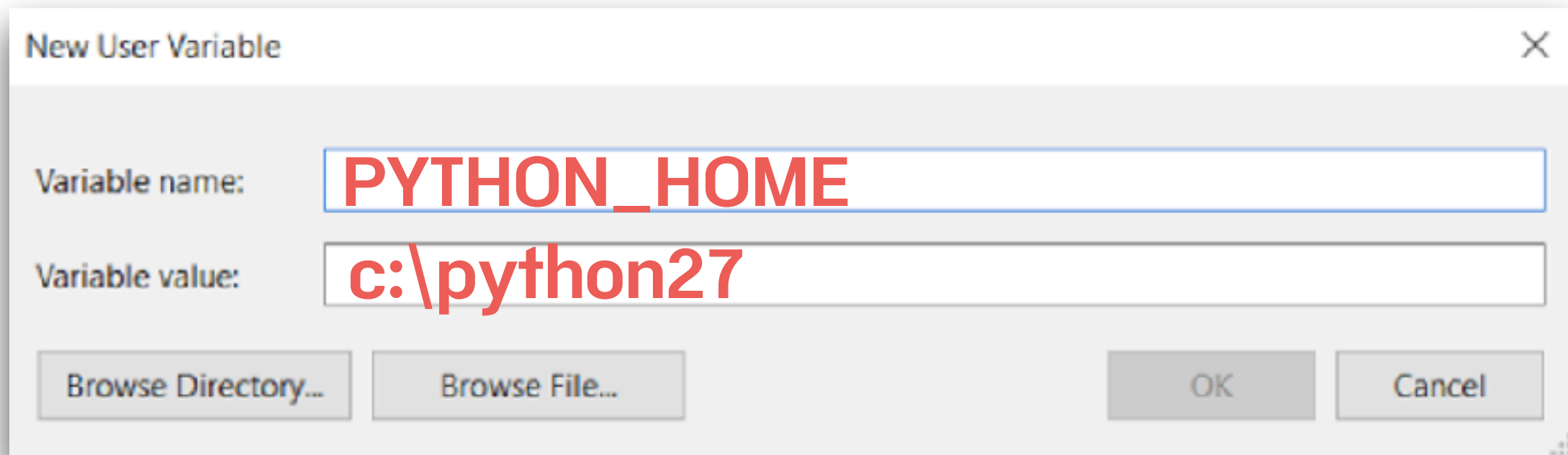
Variable name:

Variable value:

Browse Directory... Browse File... OK Cancel



Configuration PYTHON_HOME



New User Variable

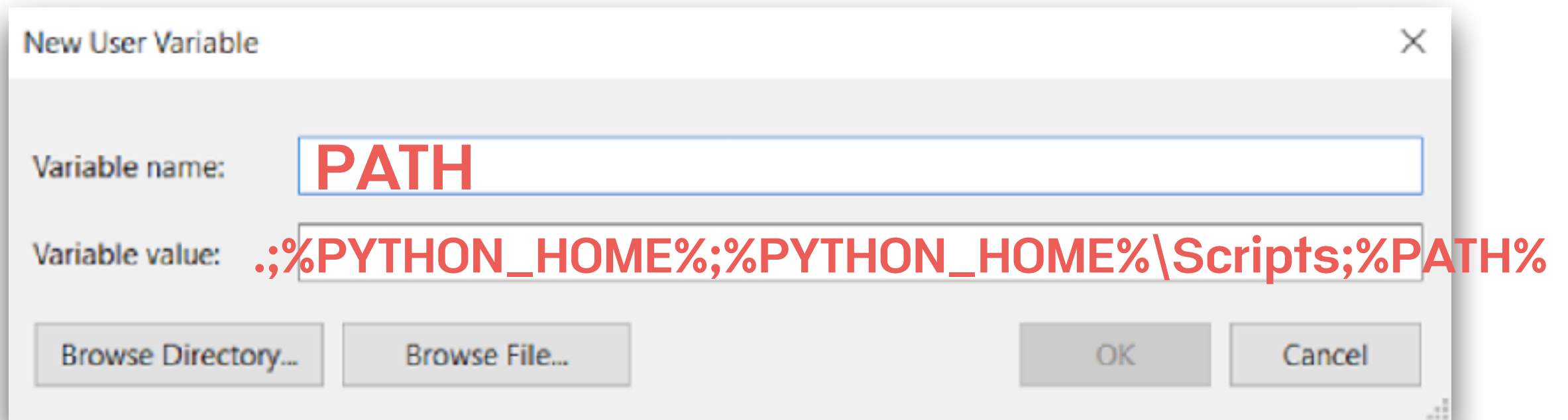
Variable name: PYTHON_HOME

Variable value: c:\python27

Browse Directory... Browse File... OK Cancel



Configuration PATH



New User Variable

Variable name: **PATH**

Variable value: **.;%PYTHON_HOME%;%PYTHON_HOME%\Scripts;%PATH%**

Browse Directory... Browse File... OK Cancel



For MacOS



Install

```
$easy_install pip
```



Install

```
$brew install robot-framework
```



Install Robot Framework



Install Robot Framework

\$pip install robotframework

```
Command Prompt

C:\Users\SCK>pip install robotframework
Collecting robotframework
  Downloading robotframework-3.0.2.tar.gz (440kB)
    100% |#####| 450kB 656kB/s
Installing collected packages: robotframework
  Running setup.py install for robotframework ... done
Successfully installed robotframework-3.0.2
```



Check Robot Framework

\$pybot

```
Command Prompt

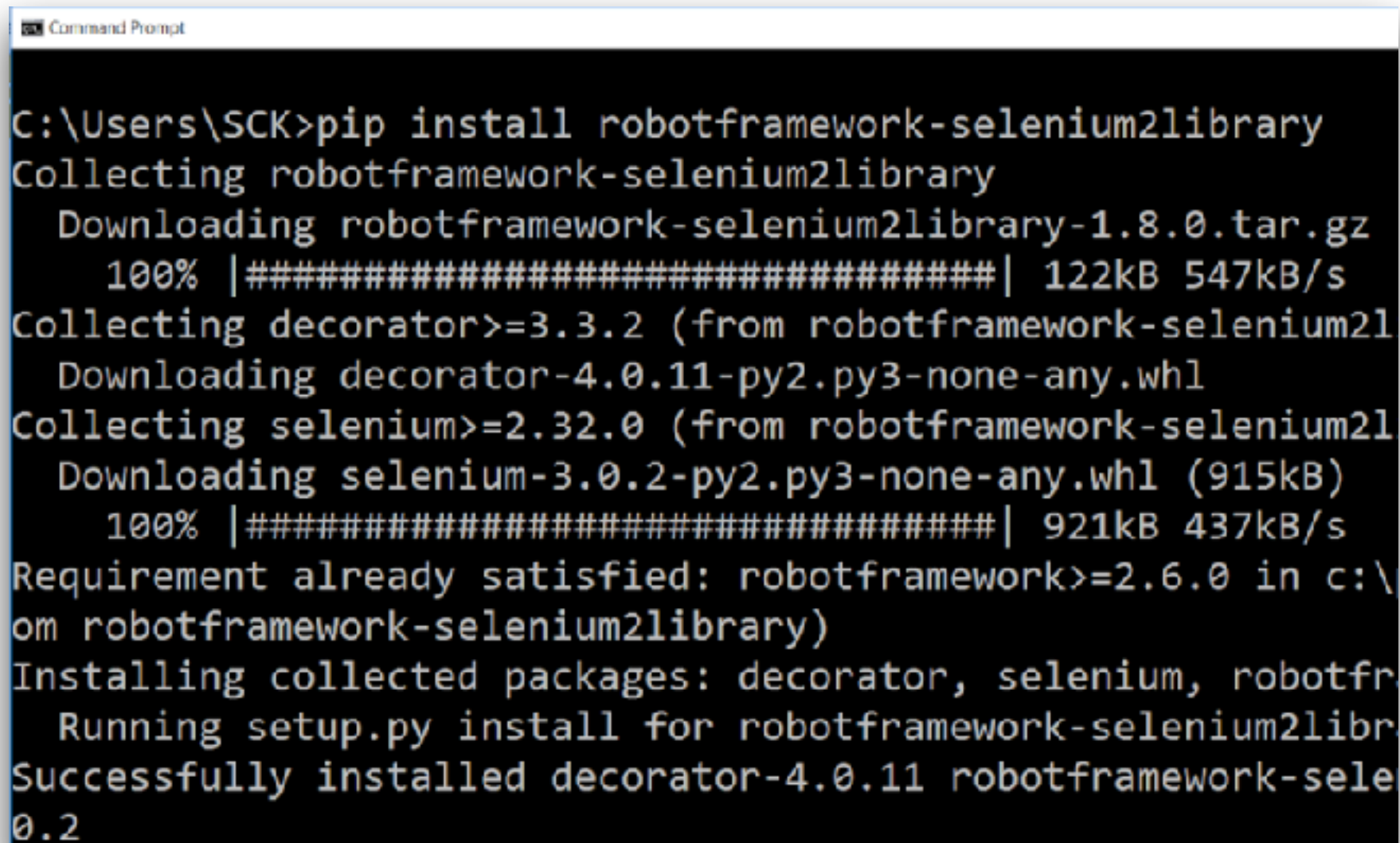
C:\Users\SCK>pybot
[ ERROR ] Expected at least 1 argument, got 0.

Try --help for usage information.
```



Install Robot Framework Library

\$pip install robotframework-selenium2library



```
Command Prompt
C:\Users\SCK>pip install robotframework-selenium2library
Collecting robotframework-selenium2library
  Downloading robotframework-selenium2library-1.8.0.tar.gz
    100% |#####| 122kB 547kB/s
Collecting decorator>=3.3.2 (from robotframework-selenium2library)
  Downloading decorator-4.0.11-py2.py3-none-any.whl
Collecting selenium>=2.32.0 (from robotframework-selenium2library)
  Downloading selenium-3.0.2-py2.py3-none-any.whl (915kB)
    100% |#####| 921kB 437kB/s
Requirement already satisfied: robotframework>=2.6.0 in c:\python27\lib\site-packages (from robotframework-selenium2library)
Installing collected packages: decorator, selenium, robotframework-selenium2library
  Running setup.py install for robotframework-selenium2library
Successfully installed decorator-4.0.11 robotframework-selenium2library-1.8.0
```



Install IDE and Editor



IDE and Editor



Let's start to coding

