

EFFECTIVE SOFTWARE DEVELOPMENT

Mark Dawson, Ed Bennett, Michele Mesiti

AIMLAC CDT Introduction, 2019-09-25

WHAT ARE THE CHALLENGES TO WRITING SOFTWARE?

- Writing software is about telling computers what to do
- Why might this be challenging for humans to do effectively?
- Discuss in your teams
- Write thoughts down on the flipchart paper
- 10 minutes

HOW CAN WE MITIGATE THESE DIFFICULTIES?

- What techniques, methods, or language features would help mitigate these difficulties?
- Discuss in your teams
- Write thoughts down on the flipchart paper
- 10 minutes

WHAT TOOLING COULD BE ADDED TO HELP WITH THESE DIFFICULTIES?

- Using your imagination invent some tools (they may exist) that could make the development of software simpler, quicker and more productive
- Discuss in your teams
- Write thoughts down on the flipchart paper
- 10 minutes

SYNTAX HIGHLIGHTING

```
1 context.py  2³  utils.py •  
12  
13 class ContextPopException(Exception):  
14     "pop() has been called more times than push()"  
15     pass  
16  
17 |  
18 class ContextDict(dict):  
19     def __init__(self, context, *args, **kwargs):  
20         super(ContextDict, self).__init__(*args, **kwargs)  
21  
22         context.dicts.append(self)  
23         self.context = context  
24  
25     def __enter__(self):  
26         return self  
27  
28     def __exit__(self, *args, **kwargs):  
29         self.context.pop()  
30
```

context.py [python][unix→utf-8]

L17/267:C0 4%

AUTO-COMPLETE

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 x = np.arange(-np.pi,np.pi, 0.05)
6 y = np.sin(x)
7 plt.plot(x,y)
8 plt.title('plot with numpy data')
9
10 df = pd.DataFrame(y, index=x)
11 df.plot(title='plot with pandas data')
12
13 np..
14     abs           instance
    absolute      instance
    absolute_import instance
    add           instance
    add_docstring   function
    add_newdoc      function
    add_newdoc_ufunc function
    add_newdocs       module
    alen           function
    all            function
```

A screenshot of a code editor window titled "test_python.py". The code contains a function definition:def function(number, kwarg=3):
 if number < 2:
 raise TypeError
 return kwarg

```
The cursor is at the end of the first line of the function body. A dropdown menu is open, listing auto-complete suggestions for "np.". The suggestions include: abs, absolute, absolute_import, add, add_docstring, add_newdoc, add_newdoc_ufunc, add_newdocs, alen, and all. Each suggestion is followed by its type: instance, instance, instance, instance, function, function, function, module, function, and function respectively. The background of the code editor is dark, and the suggestions are highlighted in a light blue color.
```

INLINE DOCUMENTATION

The screenshot shows a code editor with Python code. A tooltip is displayed over the word `castleLocation`, providing documentation for the parameter.

```
def StormCastle(castleName, castleLocation, totalTroops, weapons):
    """
    Storm a castle anywhere in the world.

    :param string castleName: The name of the castle.
    :param string|castleLocation: The location of the castle.
    :param int totalTroops: The total number of troops.
    :param list[string] weapons: A list of weapons to use.
    :return: True | False
    """

    print "Storming: " + castleName + " at " + castleLocation
    print "Total Troops: " + str(totalTroops)
    print "Weapons: "
    for weapon in weapons:
        print " * " + weapon

StormCastle(1)
```

Documentation for `StormCastle(castleName, castleLocation, totalTroops, w...)`

def
StormCastle(castleName, castleLocation, totalTroops, weapons)
Inferred type: (castleName: [unicode](#), castleLocation: [unicode](#),
totalTroops: [int](#), weapons: [list\[unicode\]](#)) -> None

Storm a castle anywhere in the world.

LITERATE PROGRAMMING

IP[y]: Notebook spectrogram Last Checkpoint: a few seconds ago (autosaved) IPython (Python 3)

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#) using windowing, to reveal the frequency content of a sound signal.

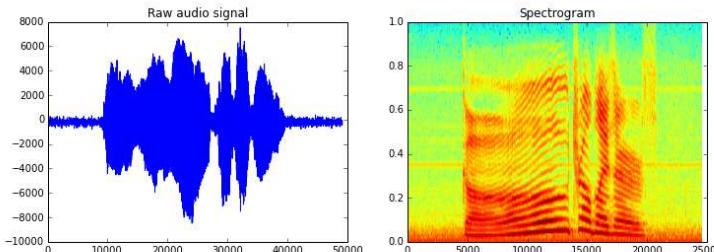
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kn} \quad k = 0, \dots, N - 1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: %matplotlib inline  
from matplotlib import pyplot as plt  
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```



VERSION CONTROL

The screenshot shows a GitHub pull request interface. At the top, it displays the repository "desktop", the branch "esc-pr" (PR #3972), and the status "Fetch origin" (last fetched 2 minutes ago). Below this, there are two tabs: "Changes" (selected) and "History". The "Changes" tab shows a list of commits:

- Appease linter
- Add event handler to dropdown component
- Move escape behavior to correct component
- Remove event handler from the branches..
- Merge branch 'master' into esc-pr
- Merge pull request #4044 from desktop/...
- Merge pull request #4070 from desktop/...
- bump to beta3
- Merge pull request #4057 from desktop/...
- Merge pull request #4067 from desktop/...
- Release to 1.1.0-beta2

Each commit includes the author's name and profile picture, the commit message, and the date it was committed. The commit "Add event handler to dropdown component" is expanded to show the diff. The diff highlights changes in the file "app\src\ui\toolbar\dropdown.tsx". The changes are as follows:

```
@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<{>
 145     this.state = { clientRect: null }
 146 }
 147
 148 + private get isOpen() {
 149 +   return this.props.dropdownState === 'open'
 150 + }
 151 +
 152     private dropdownIcon(state: DropdownState): OcticonSymbol {
 153       // @TODO: Remake triangle octicon in a 12px version,
 154       // right now it's scaled badly on normal dpi monitors.
 155
 156     @@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<{>
 249   }
 250   }
 251
 252 + private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
 253 +   if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
 254 +     event.preventDefault()
 255   }
 256 }
```

AUTOMATED TESTING

```
$ py.test -v --tb=no
----- test session starts -----
platform darwin -- Python 2.7.2 -- py-1.4.20 -- pytest-2.5.2 -- /Users/John/.pyt
honbrew/pythons/Python-2.7.2/bin/python
collected 27 items

ctest/test:0: test1 PASSED
ctest/test:12: test2 FAILED
ctest/test:16: test1 FAILED
ctest/test:0: test3 PASSED
ctest/test:0: test1 PASSED
ctest/test:0: test3 SKIPPED
ctest/test:53: test2 FAILED
ctest/test:61: test1 FAILED
ctest/test:0: test1 PASSED
ctest/test:0: test2 PASSED
ctest/test:0: test1 PASSED
ctest/test:98: test_assert_str FAILED
ctest/test:103: test_assert_equal FAILED
ctest/test:108: test_assert_not_equal FAILED
ctest/test:114: test_assert_null FAILED
ctest/test:0: test_assert_not_null_const PASSED
ctest/test:123: test_assert_not_null FAILED
ctest/test:128: test_assert_true FAILED
ctest/test:133: test_assert_false FAILED
ctest/test:0: test_skip SKIPPED
ctest/test:141: test_assert_fail FAILED
ctest/test:0: test_null_null PASSED
ctest/test:150: test_null_string FAILED
ctest/test:154: test_string_null FAILED
ctest/test:0: test_string_diff_ptrs PASSED
ctest/test:0: test_large_numbers PASSED
ctest/test:0: test_ctest_err FAILED

----- 15 failed, 10 passed, 2 skipped in 0.09 seconds -----
```

The screenshot shows a reporting interface for automated tests. At the top, it displays "62 % SW Test Academy Reporting with pytest and Tesults" and the date "Mon 9 Apr 1:11:30AM". The interface is divided into two main sections: "test_suite_a.py" and "test_suite_b.py".

test_suite_a.py:

- test3 (Module: tesults_pytest_example/test_suite_a.py) - Passed (Green)
- test2 (Module: tesults_pytest_example/test_suite_a.py) - Passed (Green)
- test1 (Module: tesults_pytest_example/test_suite_a.py) - Failed (Red)

test_suite_b.py:

- test6 (Module: tesults_pytest_example/test_suite_b.py)
 - test_input = 6*9, expected = 42 - Failed (Red)
 - test_input = 2+4, expected = 6 - Passed (Green)
 - test_input = 3+5, expected = 8 - Passed (Green)
- test5 (Module: tesults_pytest_example/test_suite_b.py) - Passed (Green)
- test4 (Module: tesults_pytest_example/test_suite_b.py) - Failed (Red)

On the right side of the interface, there are various configuration options: "VIEW: RESULTS", "PROJECT: SW Test Academy", "TARGET: Reporting with pytest and Tesults", "DATE/TIME: Mon 9 Apr 1:11:30AM", "SORT BY: Suite", "AUTO REFRESH", and a "Direct links for current selection" section with a URL.

CONTINUOUS INTEGRATION

The screenshot shows the Travis CI dashboard for the repository `green-eggs/ham`. The repository has a passing build status. The dashboard includes a sidebar with a search bar and a list of other repositories. The main area displays the current build details, commit history, and a job log.

My Repositories

- ✓ [one-fish/two-fish](#) # 2686
Duration: 33 min 46 sec
Finished: 30 minutes ago
- ✓ [hop-on/pop](#) # 7001
Duration: 22 min 54 sec
Finished: about an hour ago
- ✓ [horton-hears/awho](#) # 209
Duration: 53 sec
Finished: about 2 hours ago
- ✓ [green-eggs/ham](#) # 209
Duration: 53 sec
Finished: about 2 hours ago
- ✓ [ohthe/places-youllgo](#) # 778

green-eggs / ham build: passing

Current Branches Build History Pull Requests More options

master adding in Oh the places you'll go!

You'll be on your way up!
You'll be seeing great sights!

-o Commit abc123
Branch master
Sven Fuchs authored and committed

Ran for 53 sec
about 2 hours ago

Job log View config

Remove log Raw log

```
1 Worker information
6 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
7 Build system information
8 Build language: node_js
9 Build group: stable
10 Build dist: trusty
11 Build id: 345296935
12 Job id: 345296935
```

AUTO-REFACTORING

```
o.seek(0, 2)
total_length = o.tell()

# seek back to current position to support
# partially read file-like objects
o.seek(current_position or 0)
except (OSError, IOError):
    total_length = 0

if total_length is None:
    total_length = 0

return max(0, total_length - current_position)

def get_netrc_auth(url, raise_errors=False):
    """Returns the Requests tuple auth for a given url from netrc."""
    try:
        from netrc import netrc, NetrcParseError
        netrc_path = None
        for f in NETRC_FILES:
            try:
                loc = os.path.expanduser('~/{}'.format(f))
            except KeyError:
                # os.path.expanduser can fail when SHOME is undefined
                # getpwuid fails. See https://bugs.python.org/issue18955
                return
            if os.path.exists(loc):
                netrc_path = loc
    except:
        pass
    if netrc_path:
        return netrc(netrc_path).auth_for_url(url)
```

Change signature

```
    host_with_port == proxy_ip:
        # If no_proxy ip was defined in plain IP notation instead of CIDR
        # matches the IP of the index
        return True
    else:
        host_with_port = parsed.hostname
        if parsed.port:
            host_with_port += ':{}'.format(parsed.port)
    for host in no_proxy:
        if parsed.hostname.endswith(host) or host_with_port.endswith(host):
            # The URL does match something in no_proxy, so we don't want
            # to apply the proxies on this URL.
            return True
    with set_environ('no_proxy', no_proxy_arg):
        # parsed.hostname can be 'None' in cases such as a file URL.
        try:
            bypass = proxy_bypass(parsed.hostname)
        except (TypeError, socket.gaierror):
            bypass = False
    if bypass:
        return True
    return False
```

Extract function

```
auth.py
certs.py
compat.py
cookies.py
exceptions.py
help.py
hooks.py
models.py
packages.py
sessions.py
status_codes.py
structures.py
utils.py
```

Search Everywhere Double ⌂

Go to File ⌘O

Recent Files ⌘E

Navigation Bar ⌘P

Drop files here to open

tests

.coveragerc

.gitignore

.travis.yml

.appveyor.yml

AUTHORS.rst

DUCT.md

.md

Terminal TODO

Event Log

Git: master

Rename file/module

SYNTAX CHECKERS

```
?# encoding: utf-8
?import datetime
?from south.db import db
from south.v2 import DataMigration
?from django.db import models

!class Migration(DataMigration):

    def forwards(self, orm):
        "Write your forwards methods here."
        pass
        pass

    def backwards(self, orm):
        "Write your backwards methods here."


    models = {
        'foobar.stuffstuff': {
            'foo': {'object_name': 'bananas!'},
            'foobaz': ('django.db.models.fields.CharField', [], {'max_length': '100'}),
            'foobaz': ('django.db.models.fields.AutoField', [], {'primary_key': 'True'}),
            'foobaz': ('django.db.models.fields.CharField', [], {'max_length': '200'}),
            'foobaz': ('dianao.db.models.fields.BooleanField', [], {'default': 'False'}),
```

AUTO-FORMATTERS

```
example.py  *
1 x = { 'a':37, 'b':42,
2
3 'c':927}
4
5 y = 'hello ''world'
6 z = 'hello +'world'
7 a = 'hello {}'.format('world')
8 class foo ( object ):
9     def f (self ):
10         return 37**+2
11     def g(self, x,y=42):
12         return y
13     def f ( a ):
14         return 37++-+a[42-x : y**3]
15
```



WHAT DOES GOOD CODE LOOK LIKE?

- Each example has many positives and negatives
- Discuss in your teams. What are the good points? What are the bad ones?
- Why?
- 15 minutes

EFFECTIVE TEAM WORKING

Mark Dawson, Ed Bennett, Michele Mesiti

AIMLAC CDT Introduction, 2019-09-26

FORM YOUR TEAMS!

- No more than:
 - 1 medical student
 - 2 computer scientists
 - 3 physics & astronomy
- No more than 2 from a given institution
- Other students:
 - Computer Science, Cardiff
 - Computer Science, Bangor

SCENARIO

- You are a small software team specialising in bespoke machine learning products
- You've been hired by an (imaginary) company
- They wish to investigate automating business decisions with Artificial Intelligence, Machine Learning and/or Advanced Computing

WHAT MIGHT MAKE THIS CHALLENGING?

- What difficulties can you see with delivering software as a small team, responding to customer needs?
- Discuss in your teams
- Write thoughts down on the flipchart paper
- 10 minutes

HOW CAN TEAMS?

- Can you imagine some ways of working that might resolve these problems?
- Discuss in your teams
- Agree a series of short bullet points (max 7) describing the process you might use
- 10 minutes

SETTING UP TOOLS



Taco's Tacos ☆ | Taco & Co. Free | Team Visible | ... Show Menu

Resources

- Financials & Growth Data 0/5
- 2017 Goals And KPIs 0/2
- Brand Guide 0/1
- Employee Manual 0/1
- Add a card...

To Do

- Build A Better Burrito: 7 Layers To Success 0/7
- Nacho Ordinary Birthday - Event Space Rentals 0/0
- Taco Drone Delivery Service Nov 10 0/3
- Superbowl Ad - "Super Salad Bowls" Dec 12 0/0
- Add a card...

Doing

- The Taco Truck World Tour Oct 5 0/0
- Operation "Awesome Sauce" - A Recipe For Profit Oct 18 0/3 2/5
- #NoFiller Instagram Campaign 0/3
- Global Franchise Opportunities 0/9
- Add a card...

Done

- Focus Group: Corn vs. Flour Tortillas 0/0
- New Swag: Socks, Scarves & Salsa 0/5
- Eco Friendly Utensils & Napkins 3/3
- Update Yelp Listing 0/1
- Grand Opening Celebration Aug 11, 2016
- Add a card...



A1 Marketing ▾ Will Rodrigues

All threads

Starred

br

ev

Channels are where teams share ideas and make decisions.

Channels

accounting-costs

advertising-ops 1

annual-planning

design-feedback

marketing

urgent-issues

Direct messages

Slackbot

Elena Nowak

Matt Brewer

Toby Davies

#annual-planning

Emily Anderson 09:47
How are those year-end numbers coming along?

Paul Leung 10:18

Year-on-year growth
78 kB PDF

Sarah Parker 10:19
Wow, so much great progress! ⚡

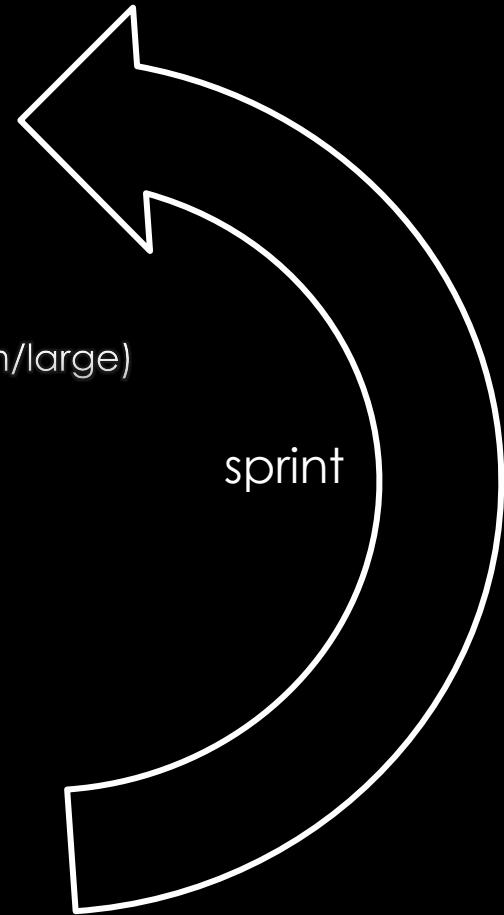
+ Message #annual-planning

@ 😊

This screenshot shows the Slack interface. On the left is a sidebar with a dark purple background. It lists channels like 'A1 Marketing' (with 1 unread message), 'Starred', and several others including '# annual-planning' (selected and highlighted in blue). Below these are 'Direct messages' and a list of users. A yellow callout box points from the sidebar towards the '# annual-planning' channel. The main area shows the '#annual-planning' channel with messages from 'Emily Anderson' and 'Paul Leung'. A file named 'Year-on-year growth' is attached. Below the channel view is a message from 'Sarah Parker' with an emoji. At the bottom is a message input field.

SAMPLE PROCESS

- Meet the customer
- Hold a planning meeting
 - Agree tasks
 - Add tasks to Trello as "waiting"
 - Estimate and note task length (small/medium/large)
 - Reflect on the last iteration
- Choose a task (waiting or review)
- Self-assign the task
- If "waiting" then move task to "active"
- Complete the task
- Move the task onwards
- Meet with your technical contact
- Have a "daily standup" at agreed intervals



DON'T BE AFRAID TO ADJUST THE PROCESS TO MAKE
IT WORK BETTER FOR YOU...!

STANDUP

- Three questions:
 - What have you done since last standup?
 - What will you work on for next standup?
 - What issues might you anticipate?



CODE REVIEW

- Why?
 - Learn
 - Share
 - Make code better

CODE REVIEW EXERCISE

- What?
 - Is it clear what the code is doing?
 - Would it be easy to reuse parts of the program? Could this fail?
 - Is it easy to test?
 - Are there any bugs?
 - Are function/variable names descriptive?
- How?
 - Be positive and constructive
 - Point out things you like
 - Clarify reasoning where code doesn't match your expectations
 - Discuss your opinions
 - Is it worth changing? Sometimes it's OK to comment but take no action

THE CODING CHALLENGE

AIMLAC CDT Introduction, 2019-09-26

THE CUSTOMER

- The customer is an energy company
- Sometimes they make oil tankers wait for favourable oil prices before docking
- Each day the tanker captain will run your system to decide if it is more profitable to dock and sell oil today or wait until tomorrow
- The current fleet of tankers each carries 750,000 barrels of West Texas Intermediate (WTI) oil
- Typical value is \$55-\$70 per barrel
- Tanker running costs are approximately \$30k/day
- The oil can be sold at the value given by the commodities market

SCENARIO

- Marketing have made initial contact with the customer, and arranged to present an initial pitch.
- In subsequent customer meetings, you'll be responsible for agreeing objectives and managing expectations

DELIVERABLE FOR FIRST CUSTOMER MEETING

- A 10 minute pitch about your proposed system
- Discuss more than one possible approach, and choose and justify recommendations
- Consider:
 - The numerical approach
 - Potential data sources
 - The advantages/disadvantages of your proposed implementation(s)
 - Interface the captain will see

FORMING YOUR TEAM

- Choose a date for your next meetings:
 - A customer meeting in approximately 2 weeks
 - A sprint planning meeting in 1 week
 - The first sprint will be shorter (around week)
- Assign a lead for this sprint (and next), who will:
 - Confirm next sprint meeting with customer
 - Chair customer meetings
 - Call a new sprint planning meeting if necessary (usually it won't be)
 - Call an extra meeting with technical contact (maximum one per sprint)
- Agree on common definition for "small", "medium" and "large" tasks
- Agree on frequency of standups
- Remember – all this can change!

HOLD YOUR FIRST PLANNING MEETING

- Discuss the problem
- Choose reasonable tasks this (short) sprint
- Agree an estimate for each task
- Put them in Trello as "waiting"
- Retrospective

HAVE FUN!