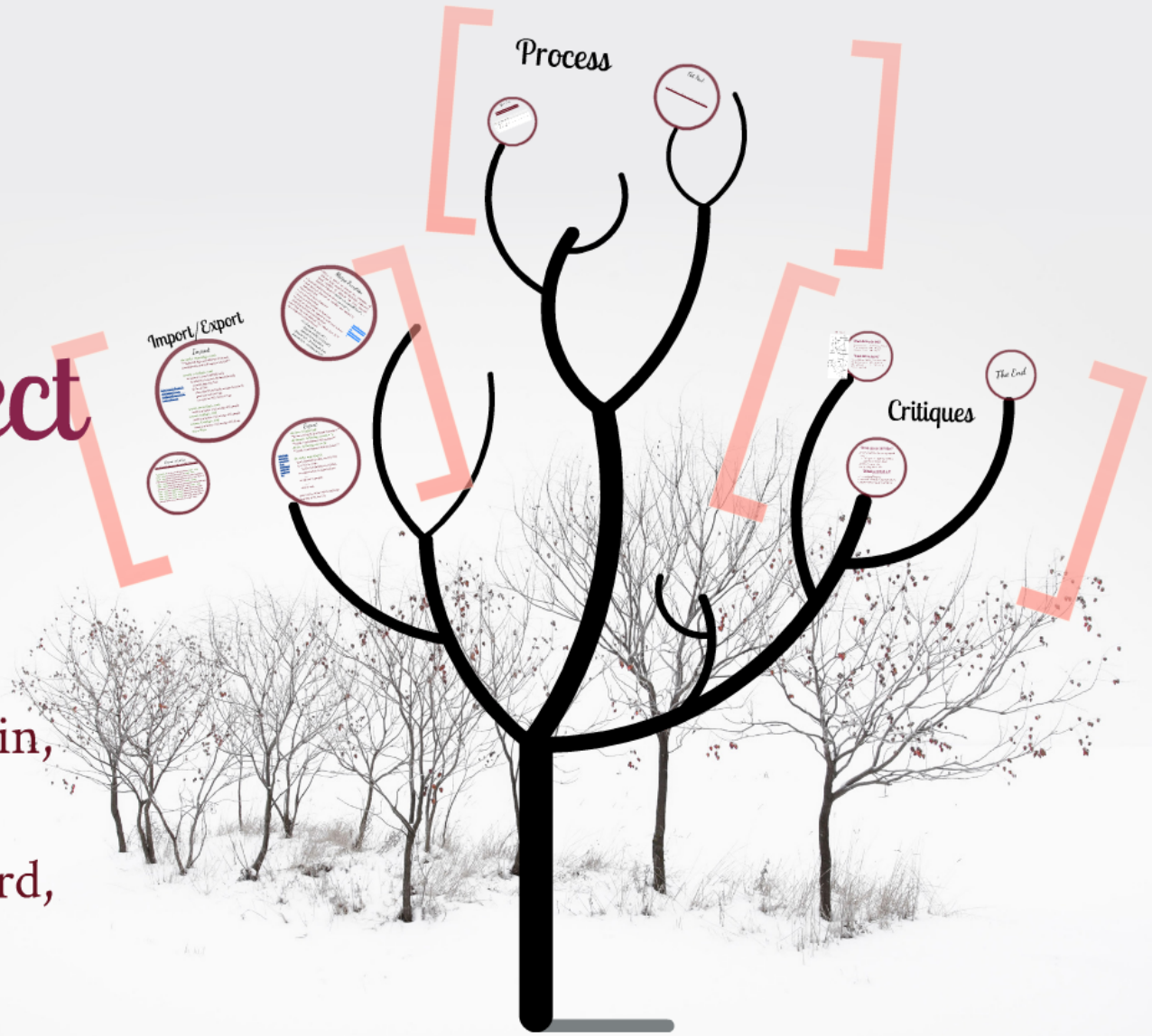
# WCDB project

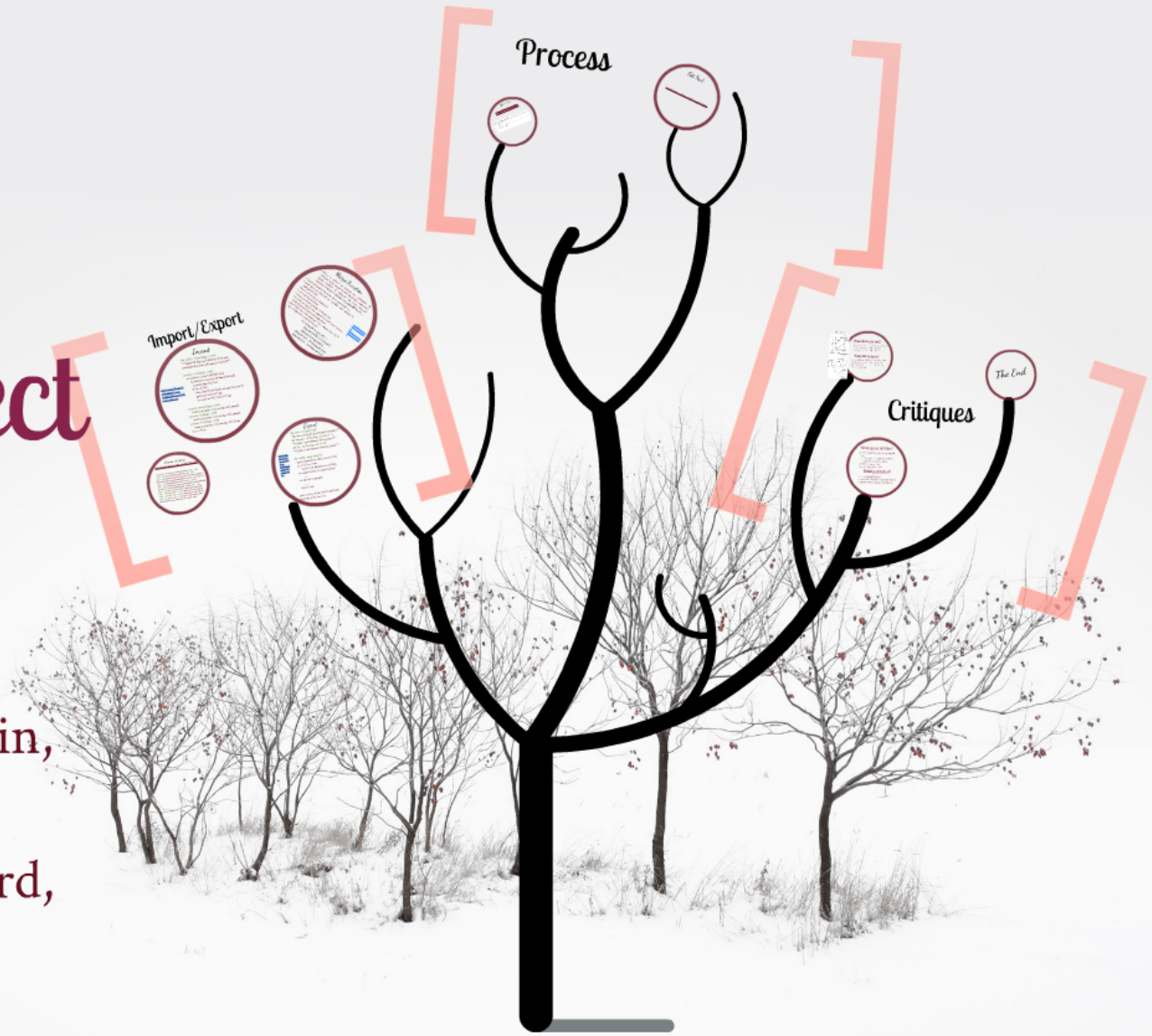
## Team Bonsai




Members: Taylor McCaslin,
Holly Hatfield,
Mallory Farr, Alex Leonard,
Wilson Bui,
Geovanni Monge, and
Daniel Ehrlich (Auditor)

# WCDB project

## Team Bonsai




Members: Taylor McCaslin,
Holly Hatfield,
Mallory Farr, Alex Leonard,
Wilson Bui,
Geovanni Monge, and
Daniel Ehrlich (Auditor)

# Import/Export

## Import

```
def wcdb3_import(login, tree):
    """Takes DB login and Element Tree and
    processes data and and imports into DB"""

    process_crisis(login, tree)
        for parent in tree.findall('Crisis'):
            for element in parent.iterdescendants():
                process tags into lists
            for len of list:
                clean data (lines breaks, escape characters)
                generate insert strings
                run queries with insert strings

    process_person(login, tree)
        same as process_crisis except with people
    process_org(login, tree)
        same as process_crisis except with people
    process_kind(login, tree)
        same as process_crisis except with kinds
    return None
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#L364

## Merge Function

- Merge is passed an element tree containing all elements from imported files (including duplicates), build variables or different top level elements (includes duplicates) cparent = tree.findall("Crisis")...
- Create list holder for ident values crises_list = []
- loop through elements in variables and append to appropriate lists
- set → set to remove duplicates
- list → list for iteration
- Iterate over lists pushing to New Root
- if ident in unique list, append to new root, remove ident from list to prevent duplication
- pass back the new Element Tree which does NOT have duplicates, return new_root

```
for element in cparent:x =
    element.attrib['crisisIdent']
    element_list.append(x)
element_list = list(element_list)
element_list= set(element_list)
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#L1

## Documentation

http://www.taylormccaslin.com/WCDB3.html

- Run WCDB3.py passes a list of file names to wcdb3_solve
- wcdb3_solve passes those file names to wcdb3_read
- wcdb3_read merges those individual files into one big element tree that contains duplicate elements (invalid xml) and returns it
- wcdb3_solve takes the tree passes it to wcdb3_merge
- wcdb3_merge takes that Tree and removes the duplicate elements and returns a duplicate free Element Tree to wcdb3_solve
- wcdb3_solve calls createDB which creates the DB and tables
- wcdb3_solve calls wcdb3_import which imports into the DB
- wcdb3_solve calls wcdb3_export which exports from the DB
- wcdb3_solve calls wcdb3_write which writes the exported Element tree to an out file using lxml pretty_print.

## Export

```
def clean_string(string):
    """given a string strip & escape characters"""
def element_builder(tag, content = ''):
    """builds 1 xml element with content"""
def attr_builder(tag, attrs = {}):
    """builds 1 xml element with attributes"""

def wcdb3_export(login):
    query database for data, store in lists
    for crisis in crises:
        build crisis elements & children
    for organization in organizations:
        .....
    for person in people:
        .....
    return root

pass root to writer which used lxml
to pretty print to a file
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#L803

# Documentation

http://www.taylormccaslin.com/WCDB3.html

- RunWCDB3.py passes a list of file names to wcdb3_solve
- wcdb3_solve passes those file names to wcdb3_read
- wcdb3_read merges those individual files into one big element tree that contains duplicate elements (invalid xml) and returns it
- wcdb3_solve takes the tree passes it to wcdb3_merge
- wcdb3_merge takes that Tree and removes the duplicate elements and returns a duplicate free Element Tree to wcdb3_solve
- wcdb3_solve calls createDB which creates the DB and tables
- wcdb3_solve calls wcdb3_import which imports into the DB
- wcdb3_solve calls wcdb3_export which exports from the DB
- wcdb3_solve calls wcdb3_write which writes the exported Element tree to an out file using lxml pretty_print.

# Merge Function

- Merge is passed an element tree containing all elements from imported files (including duplicates), build variables of different top level elements (includes duplicates) cparent = tree.findall("Crisis"),…

- Create list holder for ident values crises_list = []
- loop through elements in variables and append to appropriate lists
- list → set to remove duplicates
- set → list for iteration
- Iterate over lists pushing to New Root
- if ident in unique list, append to new root, remove ident from list to prevent duplication
- pass back the new Element Tree which does NOT have duplicates, return new_root

```
for element in cparent:x =
    element.attrib['crisisIdent']
    element_list.append(x)
element_list = list(element_list)
element_list= set(element_list)
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#l1315

# Import

```
def wcdb3_import(login, tree):
"""Takes DB login and Element Tree and
processes data and and imports into DB"""

process_crisis(login, tree)
   for parent in tree.findall('Crisis'):
       for element in parent.iterdescendants():
          process tags into lists
       for len of list:
           clean data (lines breaks, escape characters)
           generate insert strings
           run queries with insert strings

process_person(login, tree)
    same as process_crisis except with people
process_org(login, tree)
    same as process_crisis except with people
process_kind(login, tree)
    same as process_crisis except with kinds
return None
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#L364

# Export

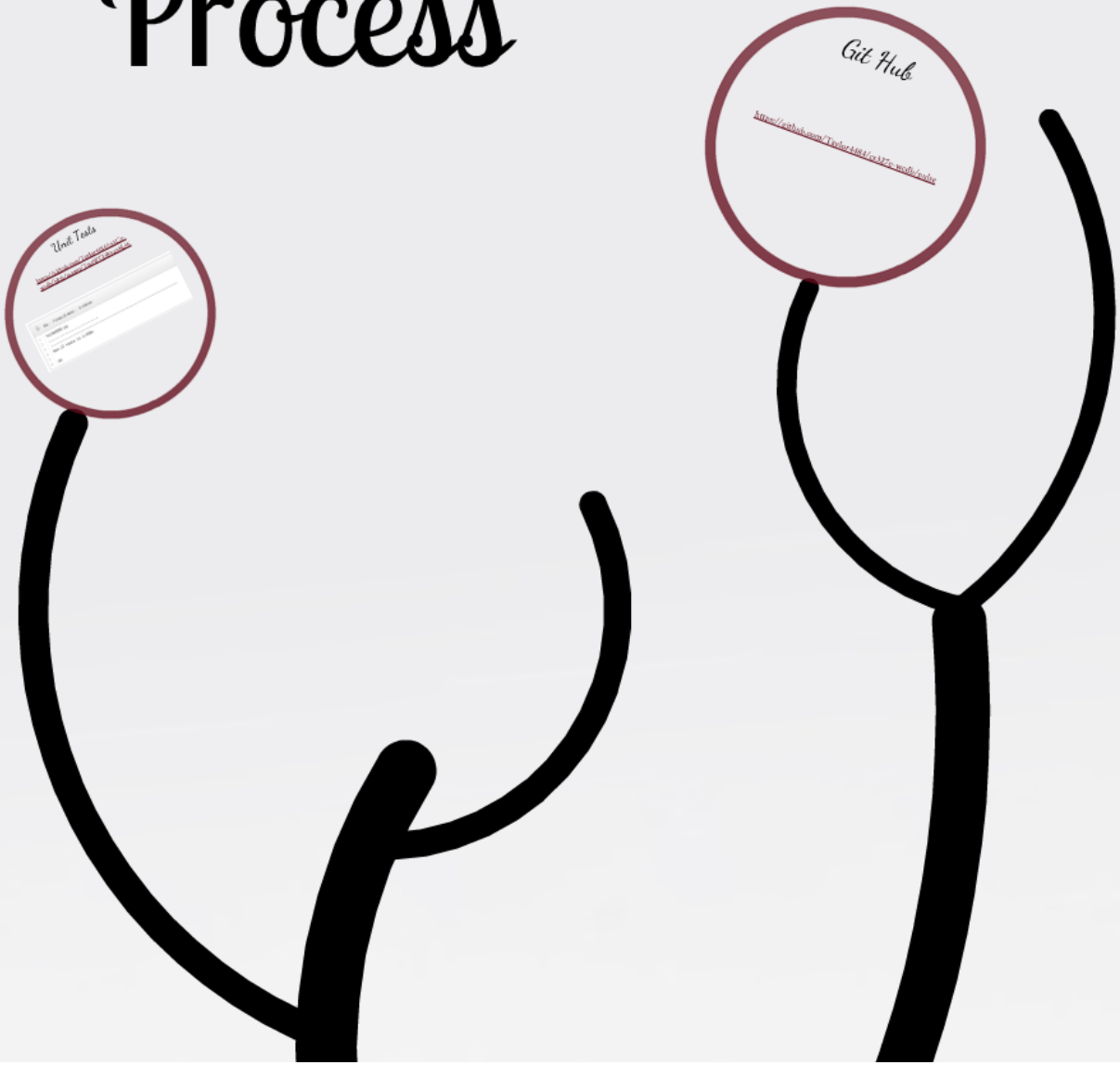
```
def clean_string(string):
"""given a string strip & escape characters"""
def element_builder(tag, content = ''):
"""builds 1 xml element with content"""
def attr_builder(tag, attrs = {}):
"""builds 1 xml element with attributes"""

def wcdb3_export(login):
  query database for data, store in lists
  for crisis in crises:
      build crisis elements & children
  for organization in organizations:
      .....
  for person in people:
      .....
  return root

pass root to writer which used lxml
to pretty print to a file
```

https://github.com/Taylor4484/cs327e-wcdb/blob/master/WCDB3.py#l803

# Process



# Unit Tests

https://github.com/Taylor4484/cs327e-wcdb/blob/master/TestWCDB3.py#L46

file | 7 lines (5 sloc) | 0.139 kb

```
TestWCDB3.py
.........................
----------------------------------------------------------------------
Ran 27 tests in 2.950s

OK
```

# Git Hub

https://github.com/Taylor4484/cs327e-wcdb/pulse

# Critiques

UML Diagram

## What did we do well?

Communication
Time management
Fixing our Errors
Python (Import/Export)
XML instances
UML Diagram

## What did we learn?

Python Modules: ..MySQL, ElementTree, XML
GitHub Functionality
Writing Shemas
The Learning Curve...
Writing Queries
Pair Programming

## What can we do better?

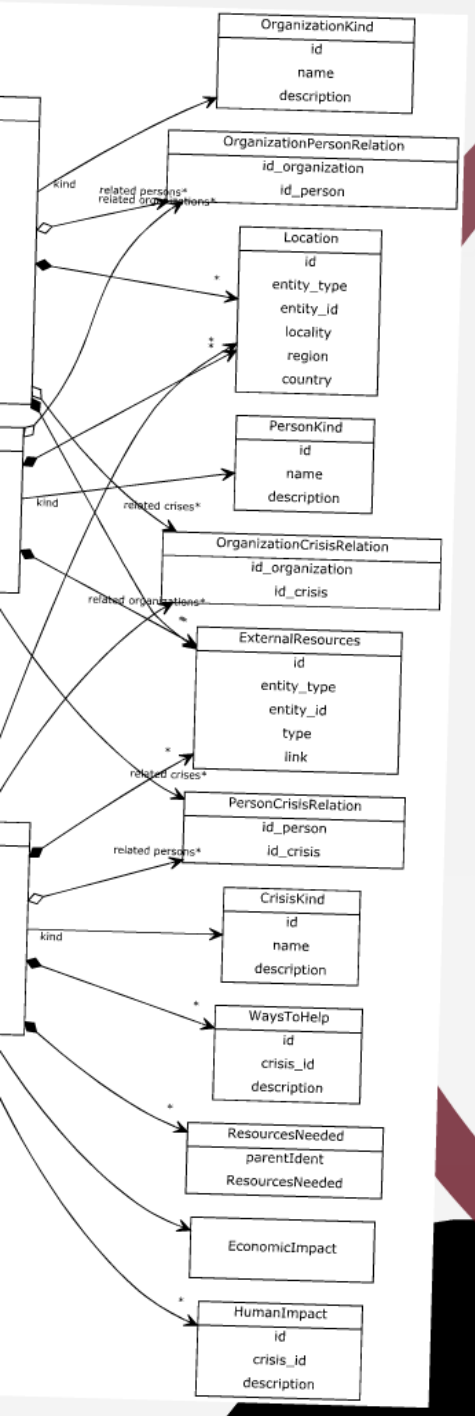
- Time management / estimating required time
  - (Let's give Dr. Downing a round of applause for the extra days!)
- More variation of tasks
- Better Delegation of tasks

## What puzzled us?

- Git Branching/Merging
- Command Line MySQL (we used Coda 2)
- Scheduling Meetings with a group of 6

The End

# UML Diagram




## What did we do well?

Communication
Time management
Fixing our Errors
Python (Import/Export)
XML instances
UML Diagram

## What did we learn?

Python Modules: _MySQL, ElementTree, XML
GitHub Functionality
Writing Queries
Writing Shemas
Pair Programming
The Learning Curve...

# What can we do better?

- Time management / estimating required time
  - (Let's give Dr. Downing a round of applause for the extra days!)
- More variation of tasks
- Better Delegation of tasks

# What puzzled us?

- Git Branching/Merging
- Command Line MySQL (we used Coda 2)
- Scheduling Meetings with a group of 6

# The End