

Evolution 4

Assessing Systems Development

By: Trevor Hofmann

What do you know?

My competencies, skills, and knowledge in terms of Systems Development would be the second best set of skills. I have been working full time in IT for right at 7 years now, since 2017 when I graduated with my bachelors in Business Administration Computer Information Systems for West Texas A&M. During my career I've not worked with any kind of Systems Development like what I've learned at school. I've architected and engineered solutions and new implementations around servers, networks, and applications like Office 365. This difference is why I feel this is my second best skill while Network & Security is my First.

While working on my graduate degree I've taken two classes around Systems Development, CIDM 6330 Software Engineering/Systems Development and CIDM 6303 Technical Foundations of CIS. These classes have been very Python heavy. I had neve coded in Python before these classes even during my undergraduate degree. From basic print statements to developing an application using Python. Here is my first Python code ever:

```
1 print("Syntax errors means there is a problem in the grammar of the expression.")
2 print(".")
3 print("An IDE is a code editor with features for autocompletion and debugging.")
4 print(".")
5 print("File extensions for a python app is .py")
6 print(".")
7 print("A linter is to check for syntax errors.")
8 print(".")
9 print("PEP8 is a style guide for best practices in Python")
10 print(".")
11 print("CPython is the default implementation of Python. ")
12 print(".")
13 print("Machine code is specific to a type of operating system/CPU.")
14 print(".")
15 print("Order of execution: Python language, cPython, bytecode, virtual machine, machine code")
16 print(".")
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE COMMENTS

CPython is the default implementation of Python.
.....
Machine code is specific to a type of operating system/CPU.
.....
Order of execution: Python language, cPython, bytecode, virtual machine, machine code
.....
[Done] exited with code=0 in 0.09 seconds

[Running] python -u "c:\Users\BlueBravo\Documents\GitHub\Masters\Portfolio\Software and Systems\6303\Python Code\assignment
Syntax errors means there is a problem in the grammar of the expression.
.....
An IDE is a code editor with features for autocompletion and debugging.
.....
File extensions for a python app is .py
.....
A linter is to check for syntax errors.
.....
PEP8 is a style guide for best practices in Python
.....
CPython is the default implementation of Python.
.....
Machine code is specific to a type of operating system/CPU.
.....
Order of execution: Python language, cPython, bytecode, virtual machine, machine code
.....
[Done] exited with code=0 in 0.068 seconds

Though out my graduate degree I spent most of the time in Python even for course that where not Systems Development related. Here is an example of my most recent python code:

```

Prototype Project Demo > Assessment Project > Project Code > src > Assessment > AssessmentAPI > views.py > customer_summary_view
1  from django.contrib.auth.models import Group, User
2  from rest_framework import viewsets
3  from django.shortcuts import render
4  from django.http import JsonResponse
5  from django.http import HttpResponse
6  import random
7  from datetime import datetime, timedelta
8  from django.db.models import Count
9  from django.shortcuts import get_object_or_404
10
11 # Importing models and serializers from the same app
12 from .models import Customer, User, Project, ProposedProject, Assessment, AssessmentTemplate, AssessmentQuestion, Recommendation, RecommendationProposedProject,
13 from .serializers import CustomerSerializer, UserSerializer, ProjectSerializer, ProposedProjectSerializer, AssessmentSerializer, AssessmentTemplateSerializer, A
14
15
16 def get_os_counts(request, customer_id):
17     # Retrieve the customer object based on the provided customer_id
18     customer = 1 # This is a placeholder value; it should be replaced with the actual retrieval logic
19
20     # Assuming TechnologyInventory model has a field 'operating_system' for the operating system
21     # Count occurrences of each operating system for the specified customer
22     os_counts = TechnologyInventory.objects.filter(customer=customer).values(
23         'operating_system').annotate(count=Count('operating_system'))
24
25     # Initialize counts for each operating system to 0
26     linux_count = 0
27     windows_count = 0
28     macos_count = 0
29
30     # Loop through the counts and update respective variables
31     for os_count in os_counts:
32         if os_count['operating_system'] == 'Linux':
33             linux_count = os_count['count']
34         elif os_count['operating_system'] == 'Windows':
35             windows_count = os_count['count']
36         elif os_count['operating_system'] == 'macOS':
37             macos_count = os_count['count']
38
39     # Return the counts as JSON response
40     return JsonResponse({
41         'linuxCount': linux_count,
42         'windowsCount': windows_count,
43         'macosCount': macos_count
44     })
45
46

```

During this time the second skill I've gained is SQL. During my undergraduate I had used SQL and now during my graduate degree. This like Python has been something I've not really used in my career. Most SQL work I do is installing it for a vendor, creating a user, or managing its backups. It is very uncommon that I have to run a query but even managing it has been made easier because I understand how it works. Here is an example of my most recent SQL work. It is a Python based SQL Script to generate data:

```

57
58 def generate_recommendations(customer_id, num_recommendations):
59     # Generate recommendation records for the given customer
60     for _ in range(num_recommendations):
61         name = f'Recommendation_{_+1}'
62         # Generate a random completion date between 6 months to 5 years from the current date
63         completion_date = datetime.now() + timedelta(days=random.randint(180, 1825))
64         # Extract only the date part and format as "YYYY-MM-DD"
65         completion_date = completion_date.strftime('%Y-%m-%d')
66         description = f'Description for {name}'
67         status = random.choice(["Completed", "In Progress", "Approved",
68                                "Declined", "On Hold", "Proposed"])
69         priority = random.randint(1, 10)
70         # Assuming there are 3 assessments per customer
71         assessment_id = random.randint(1, 3)
72         cursor.execute("INSERT INTO AssessmentAPI_recommendation (name, completion_date, description, status, priority, assessment, customer) VALUES (?, ?, ?, ?, ?, ?, ?)",
73                        (name, completion_date, description, status, priority, assessment_id, customer_id))
74         conn.commit()
75
76 # Function to generate assessments
77
78
79 def generate_assessments(customer_id, num_assessments):
80     # Generate assessment records for the given customer
81     for _ in range(num_assessments):
82         date = random_date(
83             datetime.now(), datetime.now() + timedelta(days=365))
84         score = random.randint(0, 100)
85         percent_complete = random.randint(0, 100)
86         comments = f'Comments for assessment_{_+1}'
87         status = random.choice(
88             ["Completed", "In Progress", "On Hold", "Scheduled"])
89         site_location = random.choice(["Site A", "Site B", "Site C"])
90         # Assuming there are 3 assessment templates
91         assessment_template_id = random.randint(1, 3)
92         # Assuming there are 3 alignment engineers
93         alignment_engineer_id = random.randint(1, 4)
94         cursor.execute("INSERT INTO AssessmentAPI_assessment (date, score, percent_complete, comments, status, site_location, assessment_template, alignment_engineer, customer)
95                        (date.strftime('%Y-%m-%d'), score, percent_complete, comments, status, site_location, assessment_template_id, alignment_engineer_id, customer_id))
96         conn.commit()
97
98 # Function to generate technology inventory entries
99
100

```

Python, SQL,

Where are you are weak?

Django

Where I feel my competencies, skills, and knowledge that I am least confident or proficient would center around Django as a whole. I've never used Django before my Graduate degree and I primarily only used it in CIDM-6330 which has been a recent course. My overall understanding of how it works fundamentally is much weaker than Python and SQL in general. This an example of my most Django code:

```
16
17 class UserSerializer(serializers.ModelSerializer):
18     class Meta:
19         model = User
20         fields = ("id", "name", "email_address", "security_group")
21
22 # Serializer for Project model
23
24
25 class ProjectSerializer(serializers.ModelSerializer):
26     class Meta:
27         model = Project
28         fields = ("id", "name", "status", "project_manager", "cost",
29                 "description", "customer", "completion_date")
30
31 # Serializer for ProposedProject model
32
33
34 class ProposedProjectSerializer(serializers.ModelSerializer):
35     class Meta:
36         model = ProposedProject
37         fields = ("id", "name", "completion_date", "description", "customer")
38
39 # Serializer for Assessment model
40
41
42 class AssessmentSerializer(serializers.ModelSerializer):
43     class Meta:
44         model = Assessment
45         fields = ("id", "date", "score", "percent_complete", "comments", "status",
46                 "site_location", "assessment_template", "alignment_engineer", "customer")
47
48 # Serializer for AssessmentTemplate model
49
50
51 class AssessmentTemplateSerializer(serializers.ModelSerializer):
52     class Meta:
53         model = AssessmentTemplate
54         fields = ("id", "name", "assessment_type", "description")
55
56 # Serializer for AssessmentQuestion model
57
58
59 class AssessmentQuestionSerializer(serializers.ModelSerializer):
60     class Meta:
61         model = AssessmentQuestion
62         fields = ("id", "question", "answer", "support")
63
```

Compared to everything else I've learned that was centered around Systems Development I've just spent the least amount of time with Django. It is also Python based so by nature it caused me to become better a Python while learning about it.

The Future?

For the future this one is difficult. I do not see myself becoming a database manager or a web developer. What I have received however is an understanding of how these systems work and an appreciation towards those who do it. I've just learned it is not for me. I feel like I will utilize my python skills well before I do SQL or Django. This is just caused by the power that is scripting.

Most of the scripting done during my career has been PowerShell based. I feel like this was something we missed out on during my Graduate degree. I understand it is more of a general degree, but I do feel like there were opportunities that scripting like this could have been learned in the program. I do continue to use PowerShell and will continue to utilize it in my career.