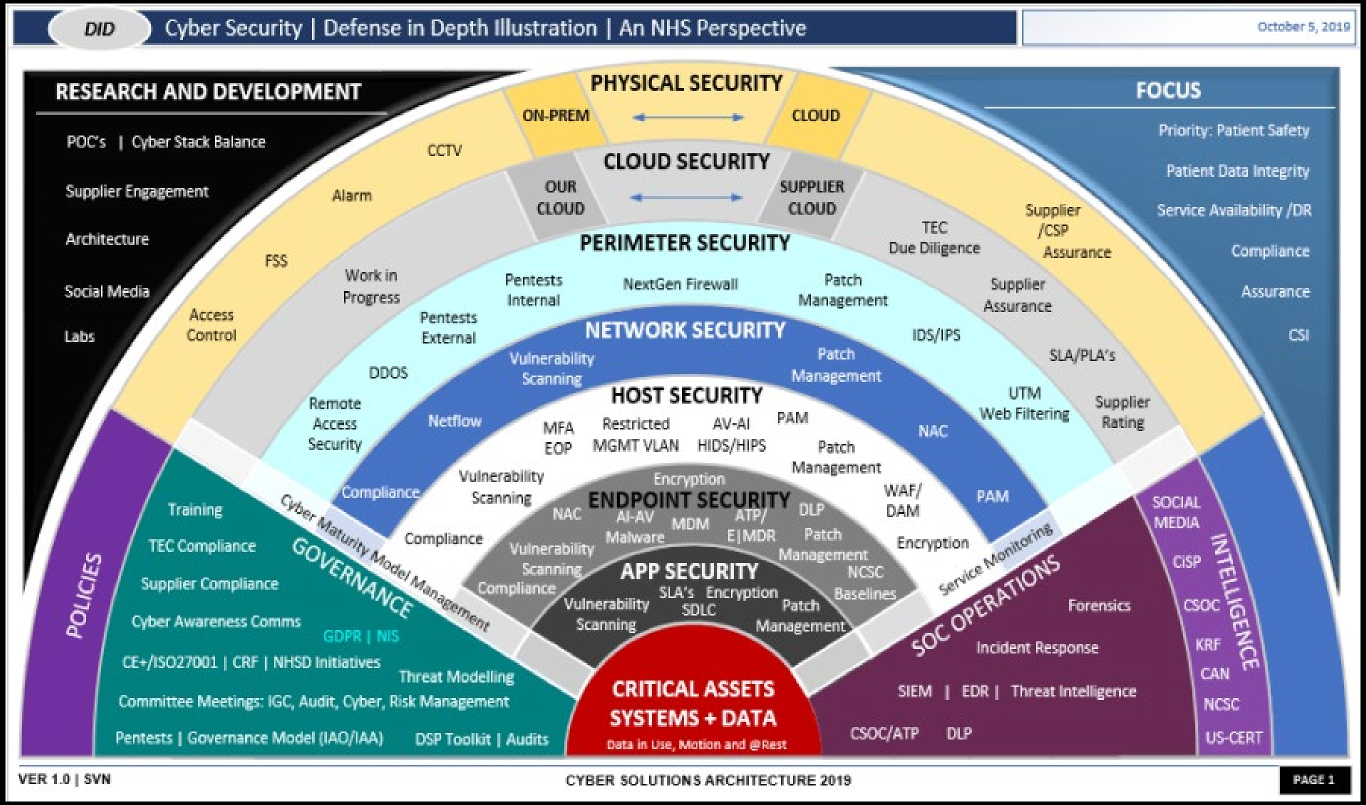
Green Pace

Security Policy Presentation

Developer: *Justin Swinney*

Green Pace

OVERVIEW: DEFENSE IN DEPTH



Green Pace

THREATS MATRIX

Likely Data Type String Correctness Smart Pointers	**Priority** SQL Injection Memory Protection Exceptions Data Value
Low priority Assertions	**Unlikely** Naming Convention Standard Library Usage

Green Pace

10 PRINCIPLES

Principles	Coding Standards
Validate Input Data	Data Type, Data Value, String Correctness, SQL Injection, Assertions, Standard Library Usage,
Practice Defense in Depth	Assertions, Exceptions,
Architect and Design for Security Policies	
Heed Compiler Warnings	Data Type,
Keep It Simple	Smart Pointers, Standard Library Usage, Naming Convention,
Adhere to the Principle of Least Privilege	
Default Deny	Data Value, SQL Injection, Memory Protection,
Sanitize Data Sent to Other Systems	SQL Injection,
Adopt a Secure Coding Standard	Data Type, SQL Injection,
Use Effective Quality Assurance Techniques	Data Type, Data Value, String Correctness, Memory Protection, Assertions, Exceptions, Smart Pointers, Naming Convention,

Green Pace

CODING STANDARDS

Label	Coding Standard	Priority
STD-002-CPP	Data Value	P18
STD-004-CPP	SQL Injection	P18
STD-005-CPP	Memory Protection	P18
STD-007-CPP	Exceptions	P8
STD-006-CPP	Assertions	P4
STD-008-CPP	Smart Pointers	P4
STD-001-CPP	Data Type	P3
STD-003-CPP	String Correctness	P2
STD-009-CPP	Standard Library Usage	P2
STD-010-CPP	Naming Convention	P2

Green Pace

ENCRYPTION POLICIES

- **Encryption in Flight –** Encryption of data transferring over the internet.
- **Encryption at Rest –** Encryption of data while stored
- **Encryption in Use –** Encryption of data while being accessed.

Green Pace

TRIPLE-A POLICIES

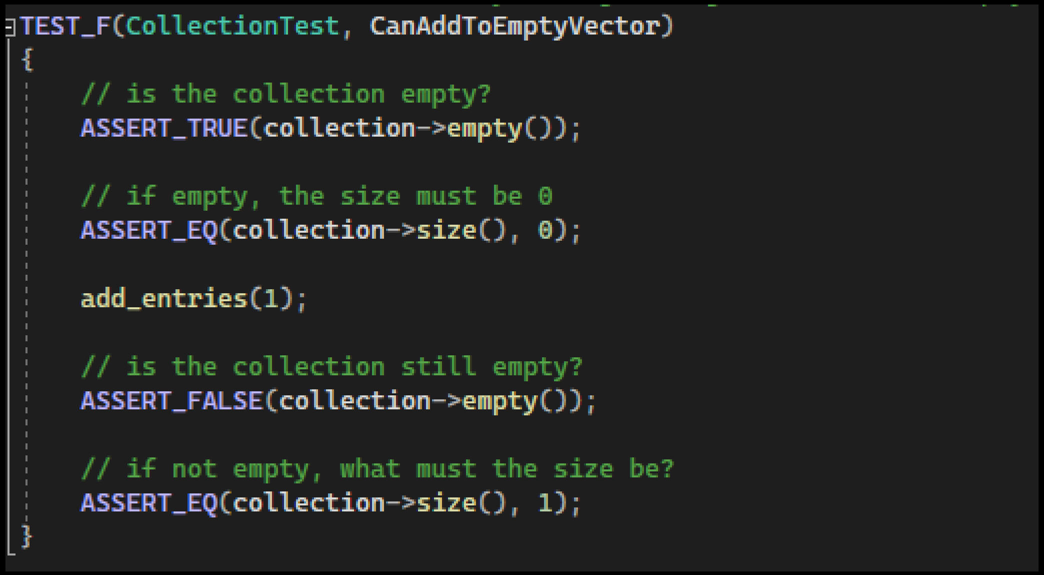
- Authentication – Verification of a user to accept or reject their request to access data or software
- Authorization – After Authentication takes place, privileges can be assigned limiting a users access to view data or utilize full functionality of the software.
- Accounting – Logging any user interaction within the software, this can be utilized during security or bug investigations.

Green Pace

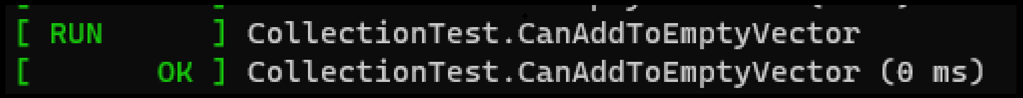
Unit Testing

What is the purpose of unit testing frameworks?

Example 1: CanAddToEmptyVector



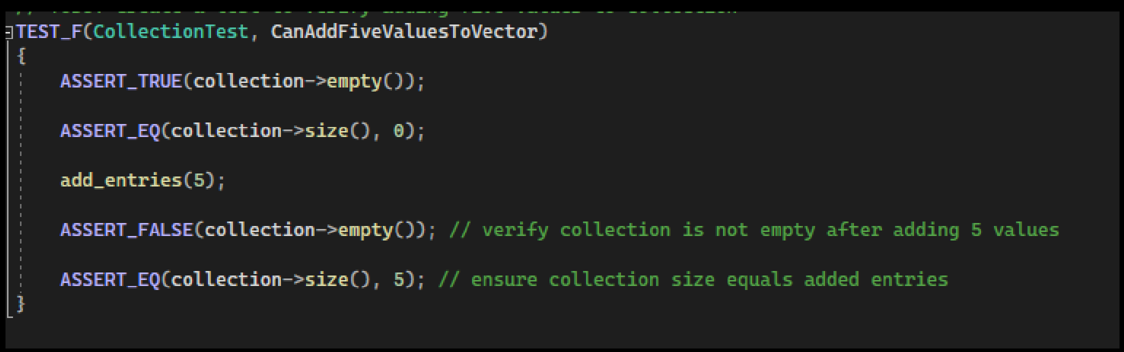
```
TEST_F(CollectionTest, CanAddToEmptyVector)
{
    // is the collection empty?
    ASSERT_TRUE(collection->empty());

    // if empty, the size must be 0
    ASSERT_EQ(collection->size(), 0);

    add_entries(1);

    // is the collection still empty?
    ASSERT_FALSE(collection->empty());

    // if not empty, what must the size be?
    ASSERT_EQ(collection->size(), 1);
}
```



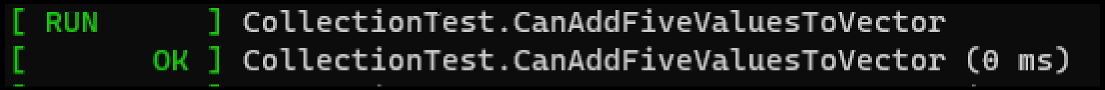
```
[ RUN      ] CollectionTest.CanAddToEmptyVector
[       OK ] CollectionTest.CanAddToEmptyVector (0 ms)
```

Green Pace

Example 2: CanAddFiveValuesToVector



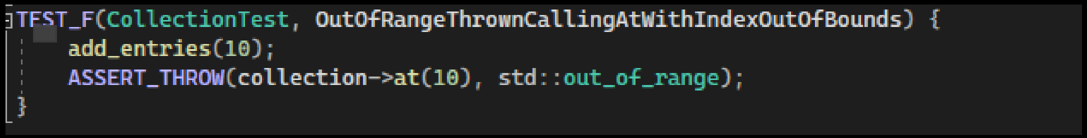
```
TEST_F(CollectionTest, CanAddFiveValuesToVector)
{
    ASSERT_TRUE(collection->empty());

    ASSERT_EQ(collection->size(), 0);

    add_entries(5);

    ASSERT_FALSE(collection->empty()); // verify collection is not empty after adding 5 values

    ASSERT_EQ(collection->size(), 5); // ensure collection size equals added entries
}
```



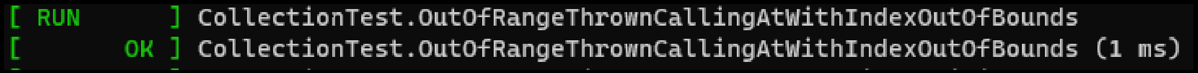
```
[ RUN      ] CollectionTest.CanAddFiveValuesToVector
[       OK ] CollectionTest.CanAddFiveValuesToVector (0 ms)
```

Green Pace

Example 3: OutOfRangeThrownCallingAtWithIndexOutOfBounds



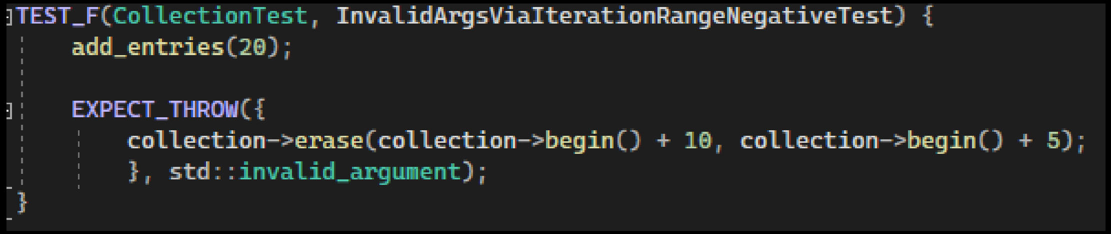
```
TEST_F(CollectionTest, OutOfRangeThrownCallingAtWithIndexOutOfBounds) {
    add_entries(10);
    ASSERT_THROW(collection->at(10), std::out_of_range);
}
```



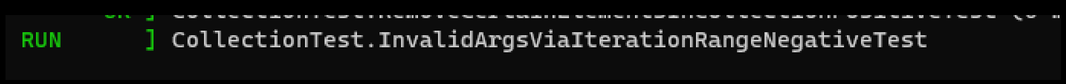
```
[ RUN      ] CollectionTest.OutOfRangeThrownCallingAtWithIndexOutOfBounds
[       OK ] CollectionTest.OutOfRangeThrownCallingAtWithIndexOutOfBounds (1 ms)
```

Green Pace

Example 3: InvalidArgsViaIterationRangeNegativeTest

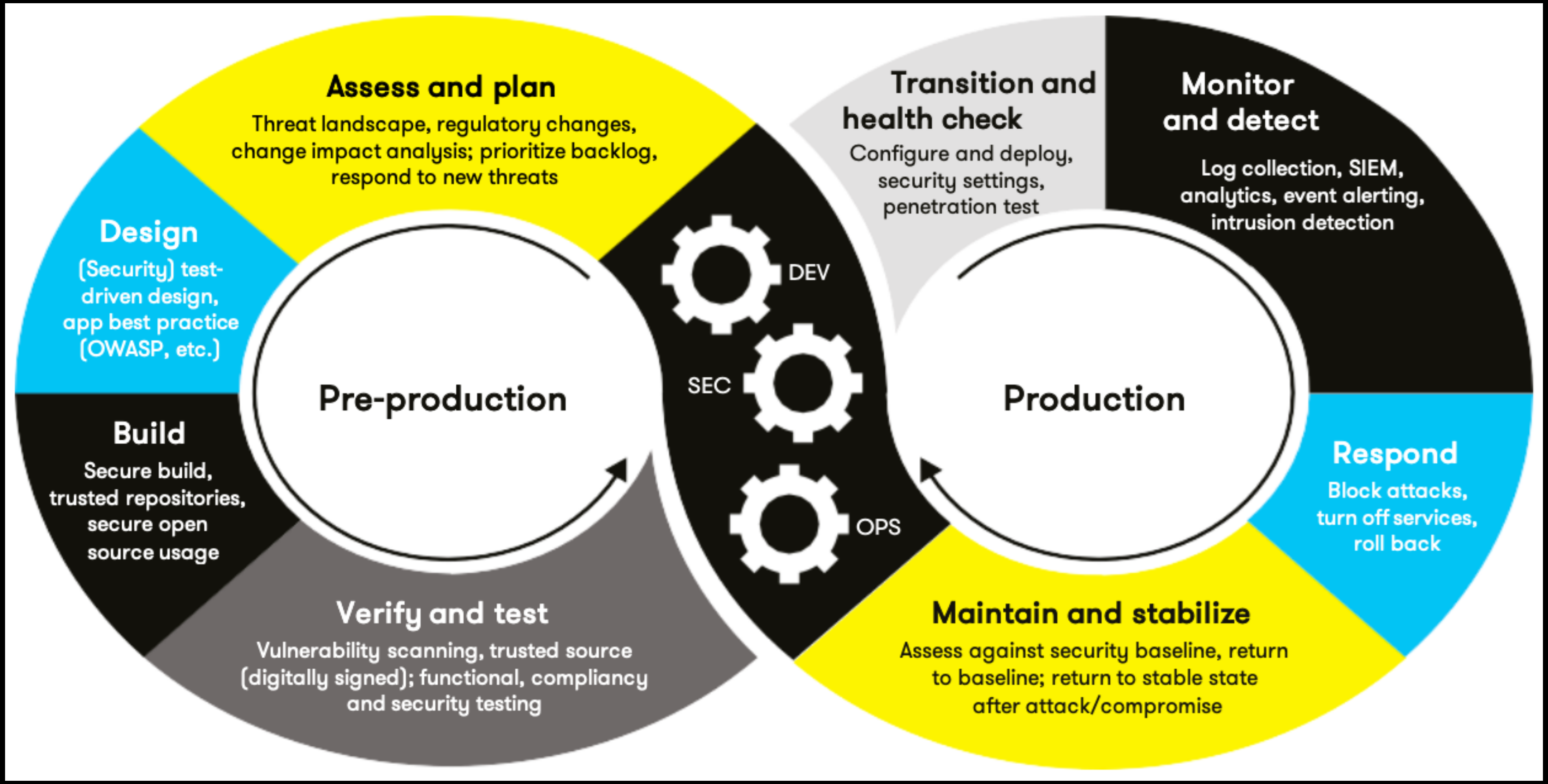


```
TEST_F(CollectionTest, InvalidArgsViaIterationRangeNegativeTest) {
    add_entries(20);

    EXPECT_THROW({
        collection->erase(collection->begin() + 10, collection->begin() + 5);
        }, std::invalid_argument);
}
```



```
[ RUN      ] CollectionTest.InvalidArgsViaIterationRangeNegativeTest
```

AUTOMATION SUMMARY


Assess and plan
Threat landscape, regulatory changes, change impact analysis; prioritize backlog, respond to new threats
Design
(Security) test-driven design, app best practice (OWASP, etc.)
Build
Secure build, trusted repositories, secure open source usage
Verify and test
Vulnerability scanning, trusted source (digitally signed); functional, compliancy and security testing
Pre-production
DEV
SEC
OPS
Transition and health check
Configure and deploy, security settings, penetration test
Monitor and detect
Log collection, SIEM, analytics, event alerting, intrusion detection
Production
Respond
Block attacks, turn off services, roll back
Maintain and stabilize
Assess against security baseline, return to baseline; return to stable state after attack/compromise

Green Pace

TOOLS

- DevSecOps pipeline is a software development strategy that utilizes industry standard security practices and combines itself into the DevOps structure.

- Pvs-studio
- CPPcheck
- Clang Tidy
- Coverity
- ParasoftJtest

Green Pace

RISKS AND BENEFITS

Problems / Solutions –

- **Third party risk** – Asses any third-party involvement with the software and ensure strict procedures are in place when your service is utilized by a third party.
- **Lacking robust coding standards** – Ensure clear and concise standards are in place with detailed threat assessments.
- **Compliance** – Ensure the software being developed meets all required compliance codes for the designated items, this should be looked at during the beginning stages of development.

Act Now

- What are the benefits?
- Are their any risk associated with acting now?

Wait

- What are the benefits of waiting?
- Are their any risk associated with waiting to act?

Green Pace

RECOMMENDATIONS

- Security Training
- Security Audits
- Automated Patch Updates.
- Enforcement of this policy

Green Pace

CONCLUSIONS

- Quality development should start with security in mind.
- Continuous improvement.
- Security starts with you.

Green Pace