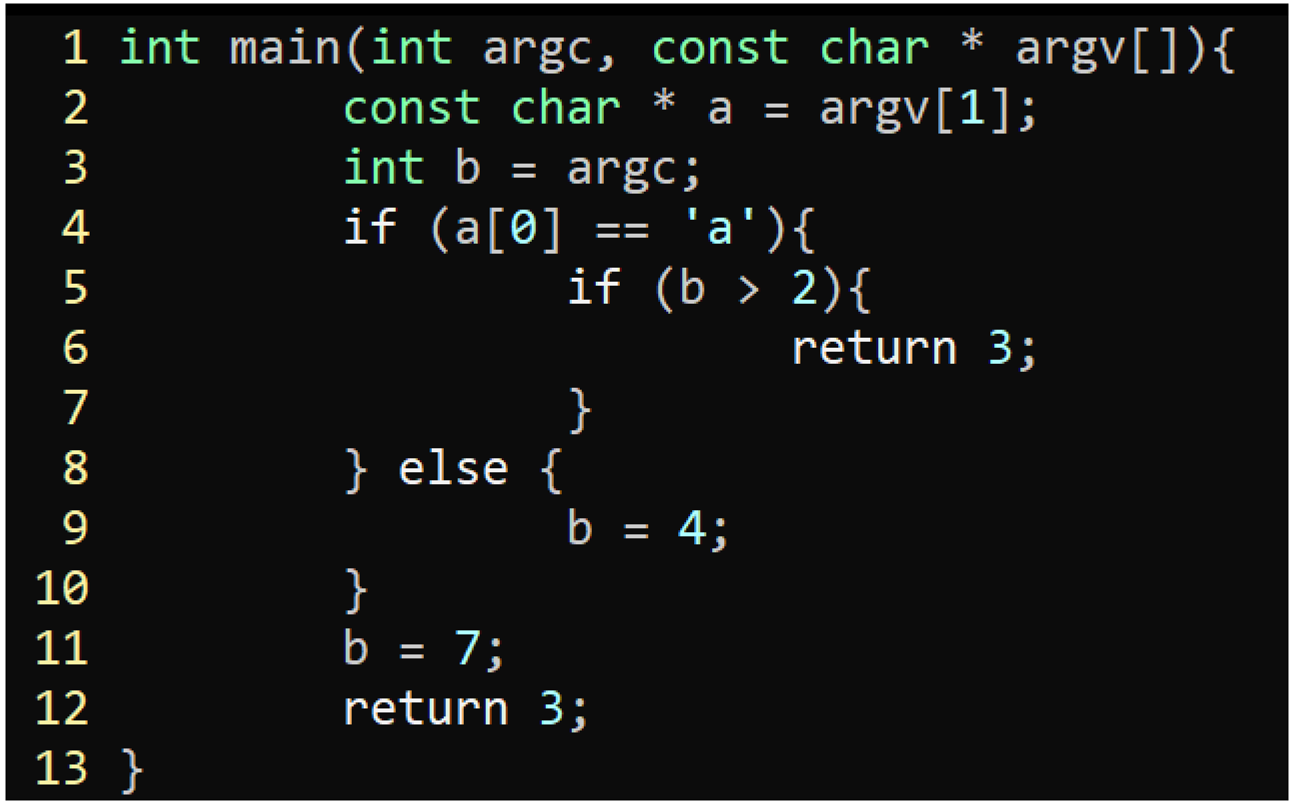
EXERCISE #23

PROGRAM SLICING REVIEW

Write your name and answer the following on a piece of paper

Draw the forward slice from line 2 in following program:



```
 1 int main(int argc, const char * argv[]){
 2     const char * a = argv[1];
 3     int b = argc;
 4     if (a[0] == 'a'){
 5         if (b > 2){
 6             return 3;
 7         }
 8     } else {
 9         b = 4;
10     }
11     b = 7;
12     return 3;
13 }
```

ADMINISTRIVIA AND ANNOUNCEMENTS

Review : 7:00 - 9:00
Wed 10/25
Pizza Provided

SSDLC

EECS 677: Software Security Evaluation

Drew Davidson

CLASS PROGRESS

WE’VE EXPLORED FORMAL TECHNIQUES TO GUARANTEE ABSENCE OF CERTAIN EXPLOITS

In practice, these tools are part of a larger discipline of secure software development

We’ll (briefly) explore some of these concepts

LAST TIME: PROGRAM SLICING

REVIEW: LAST LECTURE

EXTRACT A SUB-PROGRAM OF INTEREST BASED ON ONE (OR MORE) STATEMENTS

Forward slice

Capture all code influenced by a given statement

Backwards slide

Capture all code that influences a given statement

OVERVIEW

WE'VE SEEN THE NECESSITY OF MULTI-FUNCTION ANALYSIS IN REAL-WORLD PROGRAMS

TIME TO CONSIDER HOW IT IS DONE

LECTURE OUTLINE

- Human Factors of Security
- Security as Process
- The Secure Software Development Lifecycle

SECURING SOFTWARE IS HARD!

HUMAN FACTORS OF SECURITY

Surprising Threat Models

Security-deficient tooling

BOLT-ON SECURITY

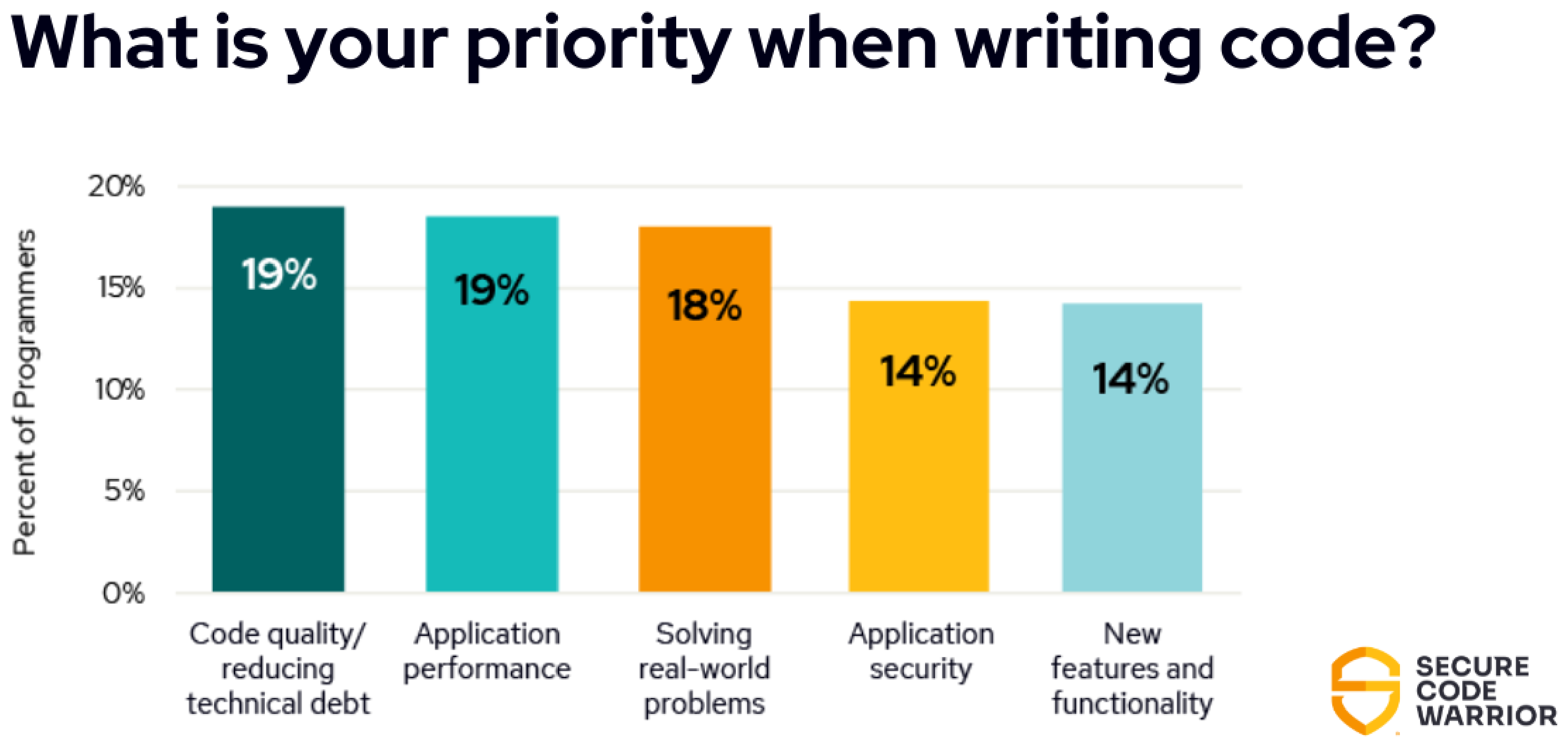
LIFECYCLES

ATTEMPTING TO RETROFIT A SECURITY SOLUTION ONTO A LEGACY SYSTEM

Sometimes necessary
Ideally avoided

VULNERABILITIES IN THE WILD

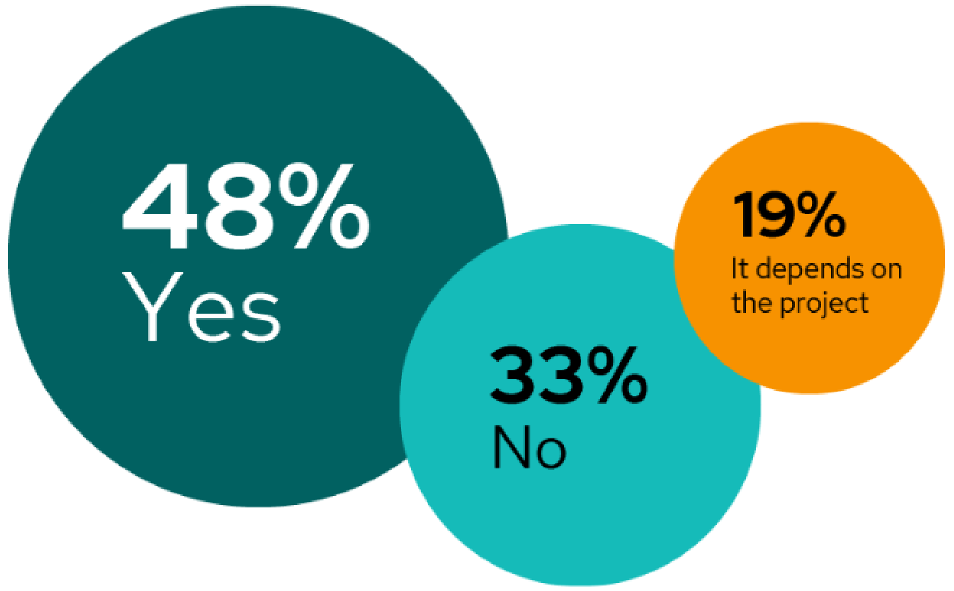
HUMAN FACTORS OF SECURITY



VULNERABILITIES IN THE WILD

HUMAN FACTORS OF SECURITY

Do you knowingly ship vulnerabilities in code?



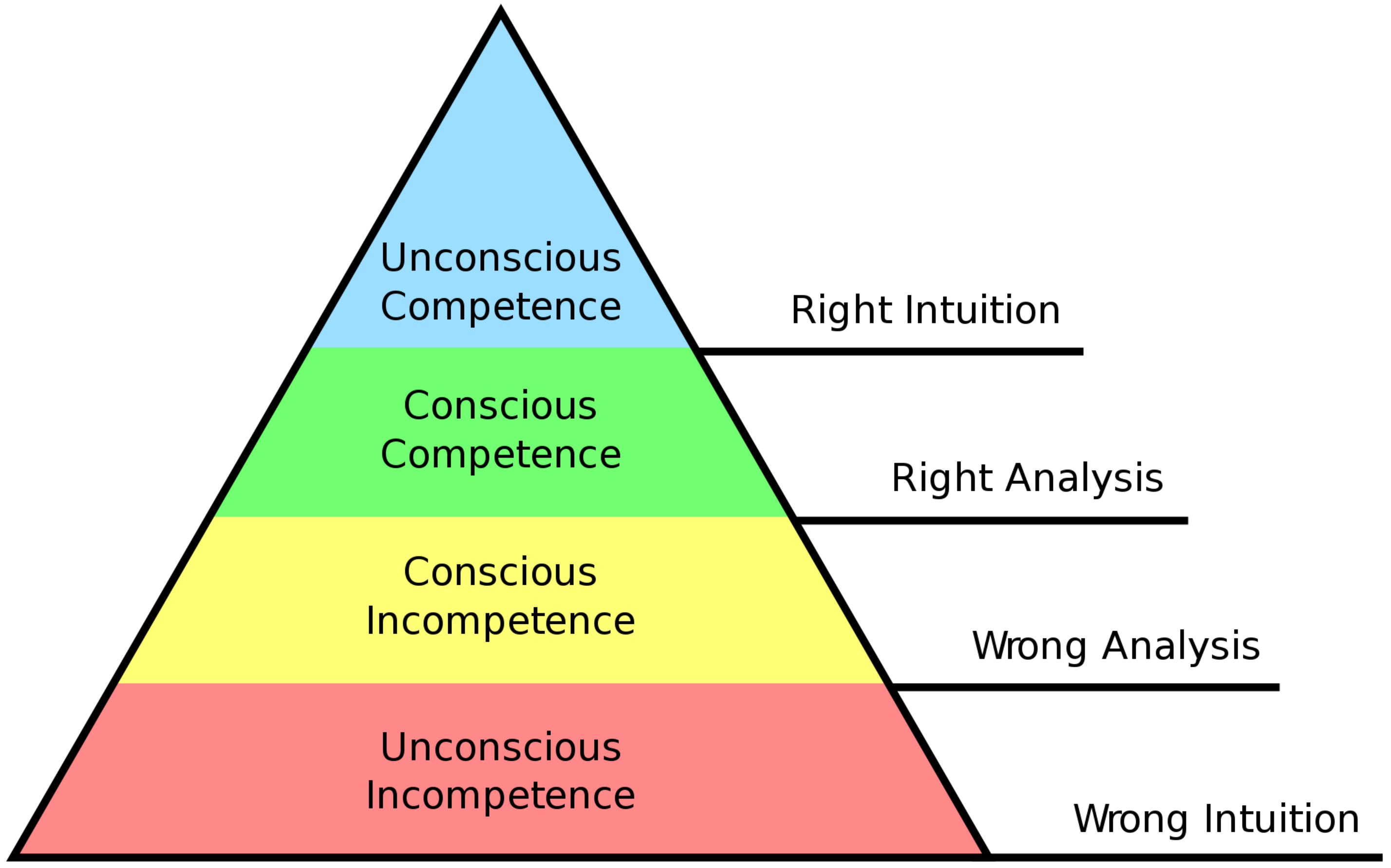
SECURE CODE WARRIOR

LECTURE OUTLINE

- Human Factors of Security
- Security as Process
- The Secure Software Development Lifecycle

PROCESS IS PROGRESS

SECURITY AS PROCESS



Hierarchy of Competence

CORPORATE SNAKE OIL

SECURITY AS PROCESS

SNAKE OIL
RELIEVES INSTANTANEOUSLY
GENUINE ARTICLE.
50c a Bottle.
SNAKE-OIL LINIMENT CURES ALL ACHES AND PAINS.
If You Are Afflicted With DEAFNESS
PURE Rattlesnake Oil
THE YAQUIS MEDICINE COMPANY

STORY TIME

SECURITY VS USABILITY

SECURITY AS PROCESS

A FUNDAMENTAL TENSION

Occurs within the implementation of software, occurs within the processes guiding software development

CONSIDER WHAT WE OWE USERS

Negative externalities

LECTURE OUTLINE

- Human Factors of Security
- Security as Process
- The Secure Software Development Lifecycle

THE “REGULAR” SDLC

LIFECYCLES

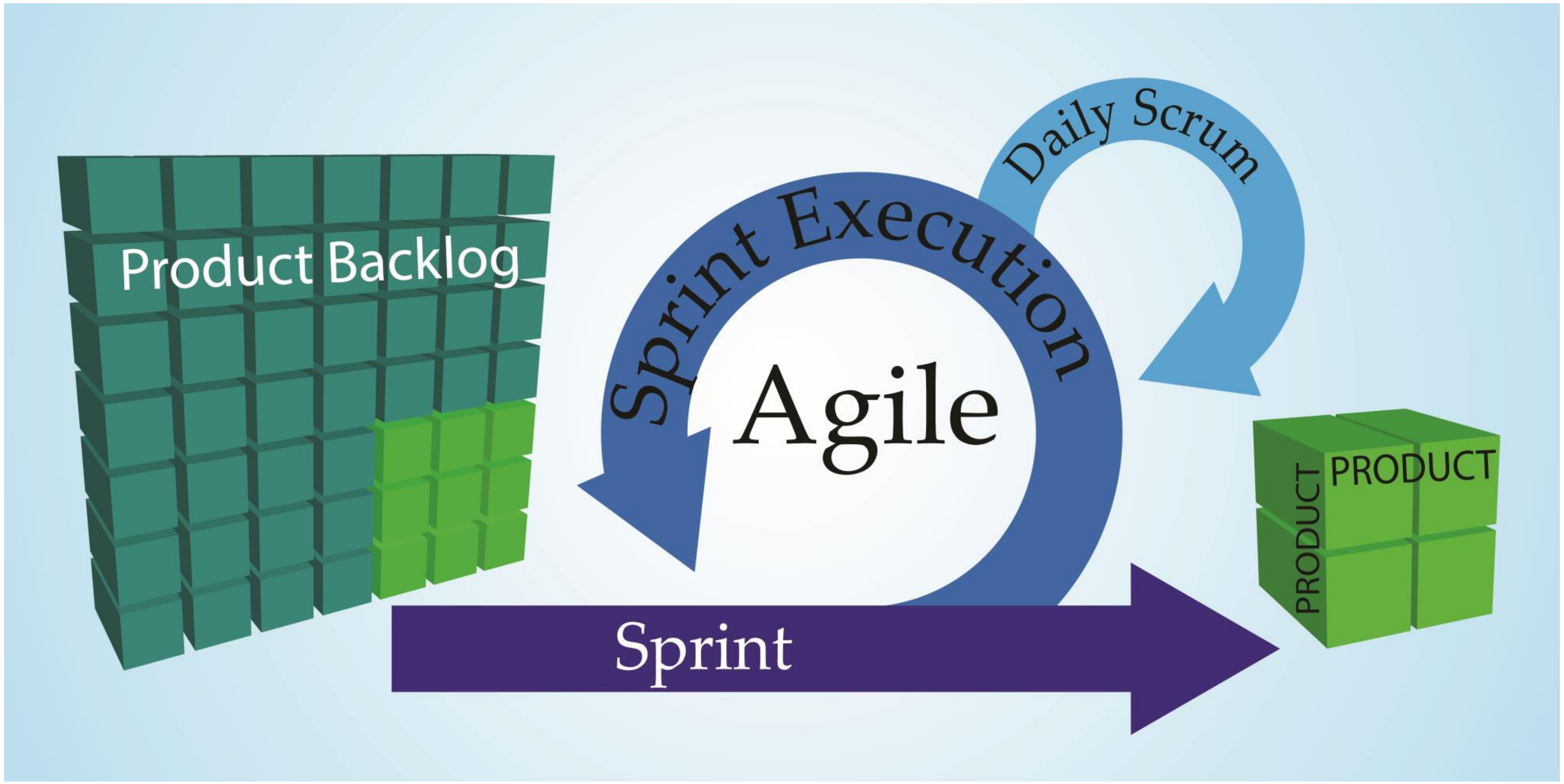
SOFTWARE DEVELOPMENT LIFE CYCLE

- Requirement analysis
- Design
- Development
- Testing and verification
- Deployment
- Maintenance and evolution

The circle of (software) life

AGILE DEVELOPMENT

SDLC: LIFECYCLES



RISK ASSESSMENT AND THREAT MODELS

THE SSDLC COMPONENTS

Companion to Requirement Phase

Functional requirement: User must verify their own contact information
Security consideration: Mechanism misuse
- Users may attempt to access the contact information of others
- Users may attempt to subvert the verification mechanism for harassment

SECURITY DESIGN REVIEW

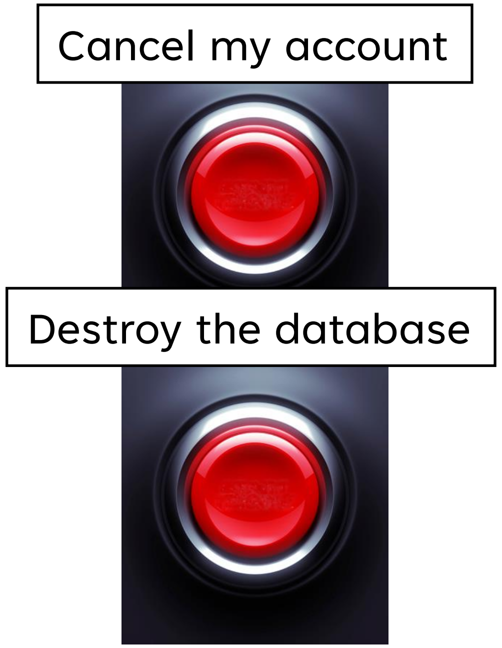
THE SSDLC COMPONENTS

Companion to the Design Phase

Functional requirement: Page should retrieve user's name, email, etc. from customer_info table in database
Security concern: Verify that user has a valid session token before retrieving information from database

Consider Security Design Principles

Principle of least privilege: Do entities in the system have exactly the privileges they need?



Bad design for a button panel

(AUTOMATED) CODE ANALYSIS

THE SSDLC COMPONENTS

Companion to Development

Apply best practices
- Accessing databases via read-only parameterized queries
- Validating user queries before processing them
- Chaos Engineering

Secure programming
- Assume a function might be misused
- Check arguments for reasonable values
- Canonicalize data

SECURITY TESTING AND CODE REVIEW

THE SSDLC COMPONENTS

Companion to Verification

Ensure proper use of APIs
- Crypto library invocations
- Holistic audits

Test the test suite:
- Evaluate the coverage of your suite
- Ensure treatment of critical functionality

Value automation:
- Repeatability / reproducibility
- Static analysis!
- Monkey testing

SECURITY ASSESSMENT AND CONFIGURATION

THE SSDLC COMPONENTS

Companion to Maintenance and Evolution

- **Logging** – Capture the behavior of the system (expected AND unexpected)
- **Metrics** – Articulate needs of the system, measure expectations against reality
- **Auditing** – Periodic retrospective analysis over codebase and configuration

WRAP-UP

- Human Factors of Security
- Security as Process
- The Secure Software Development Lifecycle