

CSE 464 Software Quality Assurance and Testing

Course Bootstrap

Xusheng Xiao

Associate Professor

School of Computing and Augmented Intelligence
Arizona State University

Instructor and Lectures

Instructor: Xusheng Xiao, xusheng.xiao@asu.com, BYENG 428

- **Research Area:** Software Engineering and Computer Security
- **Research Focus:** Mobile Security, System Security, Program Analysis, Software Analysis, Debugging
- **Office hours:** Mon-Wed 11:00am - 12:00 pm
- **Appointments:** by Zoom appointment

Classroom: Tempe – LSE106, Mon-Wed 12:00pm - 1:15 pm • **Web page:** on Canvas (<https://canvas.asu.edu/courses/231001>)

• **Slack:** <https://asu-2257-cse464-75978.slack.com/>

Homepage: <https://xusheng-xiao.github.io/>

TA and Graders

- **TA:** Liangyi Huang
 - **Email:** lhuan139@asu.edu
 - **Appointment:** Please email to set up a Zoom or in-person appointment
 - **Office hours:** BYENG 424AD, Wednesday 14:30pm-4:00pm, Friday 15:00pm-4:30pm
- **Grader:** Akilesh Shankar
 - **Email:** ashank41@asu.edu
 - **Appointment:** Please email to set up a Zoom or in-person appointment
- **UGTA:** Rudresh Bhandari
 - **Email:** rbhanda8@asu.edu
 - **Appointment:** Please email to set up a Zoom or in-person appointment
- **UGTA:** Bradley Breisch
 - **Email:** bbreisc1@asu.edu
- **TA/Grader assignments:**
 - <https://docs.google.com/spreadsheets/d/1kiC9lODzazAOubWACjqCi-MLCeR07J1NgFSnHxXin4M/edit?usp=sharing>
 - Slack/Email your assigned TA/grader first, and then escalate to instructor if needed

Textbook and Materials - 1

Title: Software Testing

Author: Jorgensen

Edition: 4th

Copyright Year: 2014

Publisher: Taylor & Francis Group,
LLC

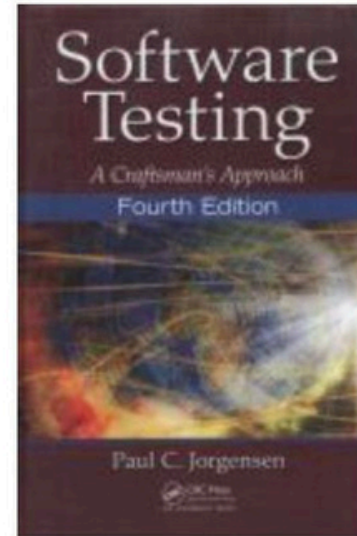
ISBN: 9781466560680

Price New: \$131.75

Price Used: \$99.00

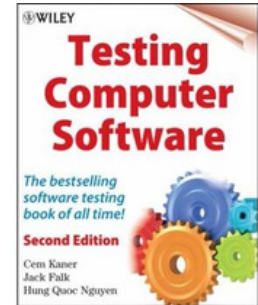
Rental Price New: \$98.81

Rental Price Used: \$52.70



Textbook and Materials - 2

Text: “Testing Computer Software”, Cem Kaner,
JackFalk, andHungQuoc Nguyen



M a t e r i a l s :

- How We Test Software at Microsoft, Alan Page;
Ken Johnston; Bj Rollison
- How Google Tests Software Book by James A.
Whittaker, Jason A. Joseph, and Jeff Carollo

ACM/IEEE Digital Library

The screenshot shows the IEEE Xplore Digital Library interface. At the top, there's a navigation bar with 'Browse', 'My Settings', 'Get Help', and 'Subscribe'. A search bar contains 'android security'. Below the search bar, it says 'Displaying results 1-25 of 1,770 for android security'. There are filters for 'Conferences (1,597)', 'Journals & Magazines (142)', 'Early Access Articles (28)', 'Books (2)', and 'Standards (1)'. A 'Year' filter is set to 'Single Year' with a range from 2002 to 2018. A 'Select All on Page' checkbox is checked. The first result is 'Research on Android Intent Security Detection Based on Machine Learning' by Lv Zhuo, Guo Zhimin, and Chen Cen, published in the 2017 4th International Conference on Information Science and Control Engineering (ICISCE). The result includes an abstract, HTML link, and a PDF icon (1361 Kb). There are also promotional banners for 'Need Full-Text' and 'MyXplore Mobile App'.

The screenshot shows the ACM Digital Library interface. At the top, there's a navigation bar with 'SIGN IN' and 'SIGN UP'. A search bar contains 'android security'. Below the search bar, it says 'Searched for android security [new search] [edit/save query]'. It also says 'Searched The ACM Full-Text Collection: 487,457 records [Expand your search to The ACM Guide to Computing Literature: 2,741,639 records]'. There are '33,653 results found'. A '434 videos found' link is visible. The first result is 'Security implications of Android: a closed system, open software mobile platform' by Hassen Saldi, published in October 2011 in the SPSM '11: Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. The publisher is ACM. Bibliometrics: Citation Count: 0, Downloads (6 Weeks): 5, Downloads (12 Months): 59, Downloads (Overall): 967. Full text available: PDF. Keywords: open software, android, security. There are also promotional banners for 'Need Full-Text' and 'MyXplore Mobile App'.

<https://dl.acm.org>

<http://ieeexplore.ieee.org/>

Google Scholar

The image shows a screenshot of the Google Scholar search results page for the query "android permissions". The search bar at the top contains the text "android permissions". Below the search bar, the results are listed. On the left side, there is a sidebar with filters. Four blue callout boxes with white text and arrows point to specific features: "Ke ywo rds" points to the search bar, "Paper (may come with pdf)" points to the first result's title, "C itat ion" points to the citation count of the first result, and "Filters" points to the left sidebar.

Google Scholar

android permissions

Articles

About 50,800 results (0.21 sec)

Any time

Since 2018

Since 2017

Since 2014

Custom range...

Sort by relevance

Sort by date

Include patents

Include citations

Create alert

Android permissions demystified

AP Felt, E Chin, S Hanna, D Song... - Proceedings of the 18th ..., 2011 - dl.acm.org

Abstract **Android** provides third-party applications with an extensive API that includes access to phone hardware, settings, and user data. Access to privacy-and security-relevant parts of the API is controlled with an install-time application permission system. We study **Android**

★ Cited by 1238 Related articles All 20 versions

Android permissions: User attention, comprehension, and behavior

AP Felt, E Ha, S Egelman, A Haney, E Chin... - Proceedings of the ..., 2012

Abstract **Android's** permission system is intended to inform users about the risks of installing applications. When a user installs an application, he or she has the opportunity to review the application's permission requests and cancel the installation if the **permissions** are

★ Cited by 690 Related articles All 22 versions

A conundrum of permissions: installing applications on an android smartphone

P Kelley, S Consolvo, L Cranor, J Jung... - ... cryptography and data ..., 2012 - Springer

Abstract. Each time a user installs an application on their **Android** phone they are presented with a full screen of information describing what access they will be granting that application. This information is intended to help them make two choices: whether or not they trust that the

★ Cited by 253 Related articles All 15 versions

Android permissions: a perspective combining risks and benefits

BP Sarma, N Li, C Gates, R Potharaju... - Proceedings of the 17th ..., 2012 - dl.acm.org

Abstract The phenomenal growth of the **Android** platform in the past few years has made it a lucrative target of malicious application (app) developers. There are numerous instances of malware apps that send premium rate SMS messages, track users' private data, or apps that,

★ Cited by 217 Related articles All 6 versions

Filters


Ke ywo rds


Paper (may come with pdf)

C itat ion


<https://scholar.google.com>

Syllabus







Account




Dashboard




Courses




Calendar




Inbox



History



Commons



Help

2023 Spring C

2023Spring-T-CSE464-32511 > Syllabus

ASU Home My ASU Colleg

63 Student View

CSE 464: Software QA and Testing (2023 Spring)

Jump to Today Edit

Course Status

Unpublish Published

Import Existing Content

Import from Commons

Choose Home Page

View Course Stream

Course Setup Checklist

New Announcement

View Course Notifications

January 2023

25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Assignments are weighted by group:

Group	Weight
Assignments	30%
Course Project	40%
Quiz and Exam	30%
Total	100%

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Quizzes

Chat

Google Drive

ASU Bookstore

Export Final Grades to PeopleSoft

MediaPlus

Zoom

Slack

Outcomes

Rubrics

Modules

Collaborations

Settings

Course and Faculty Information:

Catalog Description: Topics include software quality assurance, software quality matrices, software configuration management, software verification and validation, reviews, inspections, understanding software testing process, functional testing, structural testing, model-based testing, integration, system, and regression testing techniques, software life cycle models and software testing, testing distributed software, bug tracking, and use of testing tools.

Credits: 3

Prerequisites: CSE 360, C or better (or equivalent)

Contact Information:

- Instructor:** Xusheng Xiao
 - Office:** BYENG 428
 - Email:** xusheng.xiao@asu.edu
 - Appointment:** Please email to set up a Zoom or in-person appointment
 - Office hours:** Monday and Wednesday, 11:00am - 12:00pm, BYENG 428
- TA:** Liangyi Huang
 - Email:** lhuan139@asu.edu
 - Appointment:** Please email to set up a Zoom or in-person appointment
 - Office hours:** TBD
- Grader:** Apoorva Sritha Reddy Pindi
 - Email:** apindi@asu.edu
- Grader:** Aditi Shashank Joshi
 - Email:** ajoshi64@asu.edu
- Lecture Time/Location:** MW 3:00 pm - 4:15 pm, Tempe PSH 150
- If you have medical issues we can discuss alternative accommodations. Please communicate with me on this if you have documented medical reasons or other valid reasons to request alternative accommodations. This class is an in-person class and class attendance is required unless you have medical issues.

Note: Recordings of all class sessions will be posted in Canvas for all students to access for reviewing course materials.

Syllabus at <https://canvas.asu.edu/>

SoftwareQAand Testing



Course Structure

- Mixed with lectures and lab lectures.
 - Lectures will be recorded and uploaded to the Files section at Canvas (<https://canvas.asu.edu/courses/231001>)
 - Recorded videos will be provided before lab lectures. Students should watch the videos, practice the introduced tools, and bring questions to the lab lectures.
- Remote attendance must get approvals with valid reasons.
- Hands-on assignments and course projects
 - Assignments will need usage of commandline tools and coding
 - Course projects will produce security analysis tools
- Start installing Java 11 right away
 - <https://docs.oracle.com/en/java/javase/11/install/overview-jdk-installation.html>
- Get familiar with Bash in terminal
 - For Linux and Mac, simply open Terminal app
 - For Windows, you may install Windows Subsystem for Linux (WSL)
 - <https://learn.microsoft.com/en-us/windows/wsl/install>

Grading

- Assignments 20
- Quiz and Midterm %
- Course Project 30
- 50
- **Grade Appeal:** Please send an email to your assigned TA/grader with photos on the % grade/comment and the appeal justifications.
- **Late Submission:** All assignments are due based on the deadline set in canvas. Accommodations for late submissions with valid reasons (e.g., sick leave and family emergencies) are given only if the student communicates with the instructor **BEFORE THE DEADLINE**. Without valid reasons, late submissions are subject to penalty:
 - 1 day: 15%
 - 2-3 days: 25%
 - 4-5 days: 35%
- **Exams:** All exams are closed book and will be given in person in the classroom so that we can proctor the exams. TA and graders will take attendance for the exams. Please come 10-20 minutes earlier. Exams can be taken on Canvas or by paper.
- **Exam Conflicts :** All conflicts much be notified to the instructor **AT LEAST 3 DAYS BEFORE THE EXAM DATE**. Rescheduling for individual students is given only for valid reasons.

Academic Integrity

- Please read the academic integrity policy carefully and **STRICTLY** follow the policy.

<https://provost.asu.edu/academic-integrity>

- We spend tremendous time in grading your assignments, exams, and course projects.
- We have **NO TOLERANCE** for violations of academic integrity, e.g., cheating and plagiarizing
 - For violation of academic integrity, you will receive 0 for your work
 - We have no problem failing you in this class for the semester and having the appropriate entries placed in your ASU student records.

Generative AI

- Generative AI is a technology that can often be useful in helping students learn the theories and concepts in this course. However, unless explicitly allowed by your instructor, the use of generative AI tools to complete any portion of a course assignment or exam will be considered academic dishonesty and a violation of the [ASU Academic Integrity Policy](#). Students confirmed to be engaging in non-allowable use of generative AI will be sanctioned according to the academic integrity policy and FSE sanctioning guidelines.

GenAI Usage

- **Uses of generative AI to answer closed-book exam questions, assignments, course projects are prohibited.**
- **TA/graders will specifically check for that, and report any violations detected.**
- **Once confirmed, zero grade will be given for the exam and the student will be subject to academic integrity violation.**

Course Project

- The course project is an individual project, but will have components that require interactions with TA/grader
- The course project will include several milestones:
 1. Runnable code with unit tests
 2. Version control
 3. Code reviews
 4. Code refactorings
 5. Project report

Unit Tests

Unit tests are typically automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended.

The JUnit logo, featuring the word "JUnit" in a green and red serif font.

googletest
Google C++ Testing Framework

pytest

JUnit.net

Doubling the balance and then plus 10

```
public int calAmount () { int  
    ret = balance *3; ret=  
    ret + 10;  
    return ret;  
}
```

```
void testCalAmount() { Accounta =  
    new Account();  
    Account.setBalance(0);  
    int amount = Account.calAmount();  
    assertTrue(amount == 10);  
}
```


Version Control

- Version control, also known as source control, is the practice of tracking and managing changes to software code.
- Version control systems are software tools that help software teams manage changes to source code over time.



Code Review

Code review is a software quality assurance activity in which one or several people check a program mainly by viewing and reading parts of its source code, and they do so after implementation or as an interruption of implementation.

The screenshot displays a GitLab code review interface. On the left, a sidebar shows 'Suggested change' with a diff view for the file `lib/gitlab/no_cache_headers.rb`. The diff shows a change from `<h1>HELLO WORLD!</h1>` to `<h1>Hello World</h1>`. A 'Commit message' box is visible with the text 'Apply suggestion to %(file_path)'. The main area shows the code diff for `lib/gitlab/no_cache_headers.rb` with line numbers 4 to 11. A review comment is being written, stating 'This looks great! Just one small question about a spec and then we can get this merged! :rocket:'. The comment includes a code block with the following content:

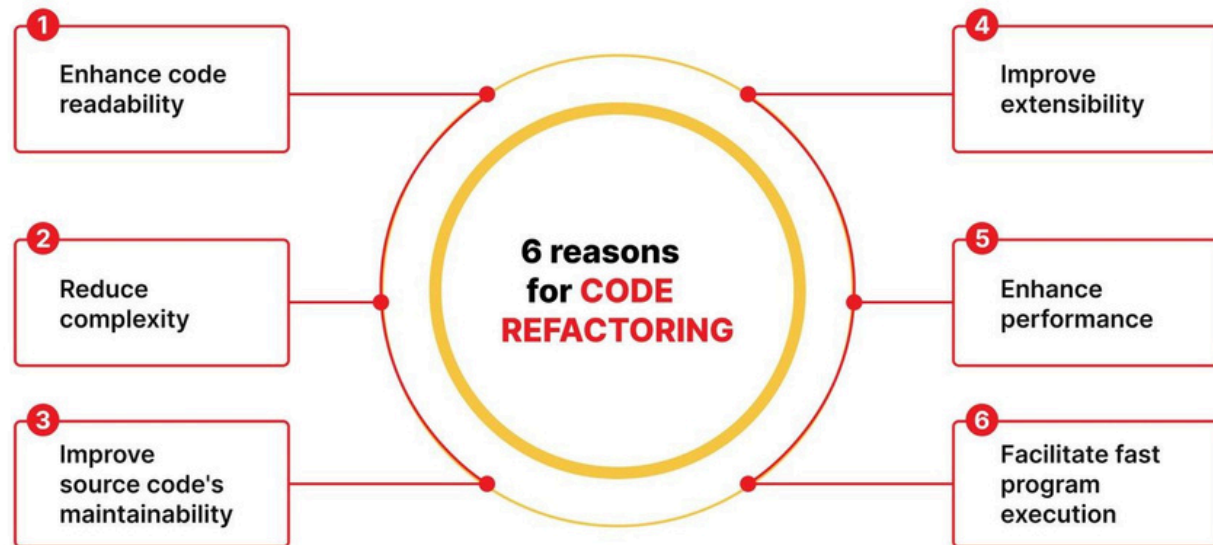
```
/unassign_reviewer @annabeldunstone
/unlabel ~"workflow::in review"
```

Below the comment, there is a checkbox for 'Approve merge request' and a 'Submit review' button. At the bottom right, there are buttons for 'Pending comments 1' and 'Finish review'.

Code Refactorings

Change the internal structure of a program without changing its external behavior

Done to make code **easier** to understand and **cheaper** to modify



ht tps: //m add evs .iolb log/ cod e-re fact orin g/

Project Report

- Describe the goal and the design of your developed software
- Provide inputs, outputs, screenshots of your running code
- Provide execution statistics of your tests
- Provide statistics of your code commits
- Provide statistics and screenshot of your code reviews
- Describe your code refactorings

Thank You !



Questions ?

Thank You !



Questions ?