Software Testing

Course Overview

Dr. Elham Mahmoudzadeh
Isfahan University of Technology
Mahmoudzadeh@iut.ac.ir
2023

*As the software industry moves into the second decade of the 21st century, **software quality** is increasingly becoming essential to all businesses and knowledge of **software testing** is becoming necessary for all software engineers.*

با ورود صنعت نرم افزار به دهه دوم قرن بیست و یکم، کیفیت نرم افزار به طور فزاینده ای برای همه مشاغل ضروری می شود و دانش تست نرم افزار برای همه مهندسان نرم افزار ضروری می شود.

Course Overview

Grading policy

- 25% on individual homework (individually solve homework assignments)
- 25% on midterm exam.
- 50% on Final exam.
- +15% on project(?, in groups two or three students)

- **Late policy:**·no credit for late work.

Course Communication

- Email: Mahmoudzadeh@iut.ac.ir
- Skype: elham.Mahmoudzadeh
- Yekta.iut.ac.ir/Software Testing/Messages

References

1- P. Ammann, J. Offutt, "Introduction to Software Testing", Cambridge University Press, 2^{nd} Edition, 2017.

- **Available at: https://cs.gmu.edu/~offutt/softwaretest/**

2- Lecture notes of Dr. P. Muller, ETH Zurich: "Software Architecture and Engineering Testing", 2018.

3- Lecture notes of Dr. Darko Marinov, University of Illinois at Urbana-Champaign: "Software Testing", 2016.

4-Lecture notes of Dr. Debra Richardson, UC Irvine: "Analysis and Testing are Creative", 2002.

Course goals

- Analysis of structure and behavior of the software.
- How to form an abstract model of the software artifact.
- Teach software engineers how to test.
- Design suitable test cases.
- Become a good tester.

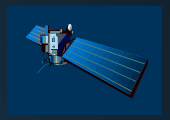
Advice

- Don’t get behind: first week especially is very fast!·
- Attend lectures: material is not all in textbook.
- Do the readings on time.

Software is a Skin that Surrounds Our Civilization

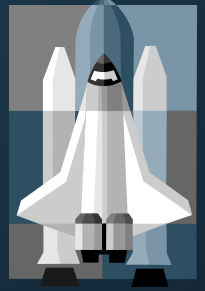
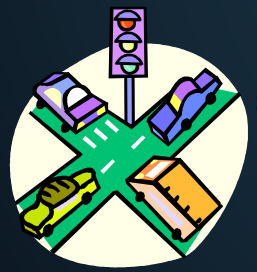










Quote due to Dr. Mark Harman

Testing in the 21st Century

- Software defines behavior
  - network routers, finance, switching networks, other infrastructure
- Today's software market :
  - is much bigger
  - is more competitive
  - has more users
- Embedded Control Applications
  - airplanes, air traffic control
  - spaceships
  - watches
  - ovens
  - remote controllers

– PDAs
– memory seats
– DVD players
– garage door openers
– cell phones

- Agile processes put increased pressure on testers
  - Programmers must unit test – with no training or education!
  - Tests are key to functional requirements – but who builds those tests ?

Industry is going through a revolution in what testing means to the success of software products

This slide is brought from "Introduction to Software Testing", Paul Ammann & Jeff Offutt, 2015.

Testing in the 21st Century

- More safety critical, real-time software
- Embedded software is ubiquitous … check your pockets
- Enterprise applications means bigger programs, more users
- Paradoxically, free software increases our expectations !
- Security is now all about software faults
  - Secure software is reliable software
- The web offers a new deployment platform
  - Very competitive and very available to more users
  - Web apps are distributed
  - Web apps must be highly reliable

بیشتر نرم افزار تعبیه شده در همه جا وجود دارد ... جیب های خود را بررسی کنید.
برنامه های کاربردی سازمانی به معنای برنامه های بزرگتر، کاربران بیشتر است
به طرز متناقضی، نرم افزار رایگان انتظارات ما را افزایش می دهد!

وب یک پلت فرم استقرار جدید ارائه می دهد
بسیار رقابتی و بسیار در دسترس برای کاربران بیشتر
- برنامه های وب توزیع می شوند
– برنامه های وب باید بسیار قابل اعتماد باشند

Industry desperately needs our inventions !

It is easy to write a program,
but
it is difficult to write a correct and reliable program.

Some Costly Failures

NASA Mars space missions

- Priority inversion (2004)
- Different metric systems (1999)

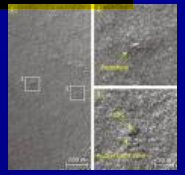
BMW airbag problems (1999)

- Recall of 15,000+ cars

- **Ariane 5 crash (1996)**
  - Uncaught exception of numerical overflow

Spectacular Software Failures

- NASA's Mars lander: September 1999, crashed due to a units integration fault



Mars Polar Lander crash site?

THERAC-25 design

- THERAC-25 radiation machine : Poor testing of safety-critical software can cost *lives* : 3 patients were killed
- Ariane 5 explosion : Millions of $$

Intel's Pentium FDIV fault(an early alarm of the need for better testing) : Public relations nightmare

Ariane 5: exception-handling bug : forced self destruct on maiden flight (64-bit to 16-bit conversion: about 370 million $ lost)

We need our software to be dependable

Testing is *one* way to assess dependability

Northeast Blackout of 2003

508 generating units and 256 power plants shut down

Affected 10 million people in Ontario, Canada

Affected 40 million people in 8 US states

Financial losses of

$6 Billion USD

The alarm system in the energy management system failed due to a software error and operators were not informed of the power overload in the system

Costly Software Failures

- NIST report, "The Economic Impacts of Inadequate Infrastructure for Software Testing" (2002)
  - Inadequate software testing costs the US alone between $22 and $59 billion annually
  - Better approaches could cut this amount in half
- Huge losses due to web application failures
  - Financial services : $6.5 million per hour (just in USA!)
  - Credit card sales applications : $2.4 million per hour (in USA)
- In Dec 2006, *amazon.com's* BOGO offer turned into a double discount
- 2007 : Symantec says that most security vulnerabilities are due to faulty software

اثرات اقتصادی زیرساخت ناکافی برای تست نرم افزار

زیان پولی در سراسر جهان به دلیل نرم افزار ضعیف خیره کننده است

زیان پولی در سراسر جهان به دلیل نرم افزار ضعیف خیره کننده است

World-wide monetary loss due to poor software is staggering

“Even when you think you’ve tested everything that you can possibly imagine, you’re wrong”

Glenn E. Reeves
(Pathfinder’s Software Team Leader)

Goal of Testing(I)

- Find faults (“Debug” Testing): a test is successful if the program fails
- Provide confidence (Acceptance Testing)
  - of reliability
  - of (probable) correctness
  - of detection of particular faults

هدف از تست
عیوب را پیدا کنید (تست »اشکالزدایی«): در صورت شکست برنامه، آزمایش موفقیتآمیز است. ارائه اطمینان (تست پذیرش)
از قابلیت اطمینان
از صحت (احتمالی)
تشخیص عیوب خاص

خطا عبارت است از انحراف رفتار مشاهده شده از رفتار مورد نیاز (مطلوب).

Goal of Testing(II)

- An error is a deviation of the observed behavior from the required (desired) behavior
  - Functional requirements (e.g., user- acceptance testing)
  - Nonfunctional requirements (e.g., performance testing)

تست فرآیند اجرای یک برنامه با هدف یافتن خطا است.

- Testing is a process of executing a program with the intent of finding an error

الزامات عملکردی (به عنوان مثال، آزمایش پذیرش کاربر)
- الزامات غیر کاربردی (به عنوان مثال، تست عملکرد)

- A successful test is a test that finds errors

This slide is brought from “Software Architecture and Engineering Testing”, Dr. P. Muller, ETH Zurich: 2018.

Why does Software contains Bugs?

- Our ability to predict the behavior of our implementations is limited
  - Software is extremely complex
  - No developer can understand the whole system

- We make mistakes
  - Unclear requirements, miscommunication
  - Wrong assumptions (e.g., behavior of operating system)
  - Design errors (e.g., capacity of data structure too small)
  - Coding errors (e.g., wrong loop condition)

چرا نرم افزار حاوی اشکال است؟
- توانایی ما برای پیش بینی رفتار پیاده سازی هایمان محدود است
- نرم افزار بسیار پیچیده است
- هیچ توسعه دهنده ای نمی تواند کل سیستم را درک کند
- ما اشتباه می کنیم
- الزامات نامشخص، عدم ارتباط

- مفروضات اشتباه (به عنوان مثال، رفتار سیستم عامل)
- خطاهای طراحی (به عنوان مثال، ظرفیت ساختار داده بسیار کم)
- خطاهای کدنویسی (به عنوان مثال، وضعیت حلقه اشتباه)

This slide is brought from “Software Architecture and Engineering Testing”, Dr. P. Muller, ETH Z

محدودیت های تست
"تست برنامه را می توان برای نشان دادن وجود اشکالات استفاده کرد، اما هرگز برای نشان دادن عدم وجود آنها نمیتوان استفاده کرد."

Limitations of Testing

"Program **testing**
can be used to **show the presence of bugs**,
but **never** to **show their absence**!"

[E. W. Dijkstra]

- It is **impossible** to **completely** test any nontrivial module or any system
  - **Theoretical** limitations: **termination**
  - **Practical** limitations: prohibitive in **time and cost**

آزمایش کامل هر ماژول غیر ضروری یا هر سیستمی غیرممکن است
- محدودیت های نظری: خاتمه
- محدودیت های عملی: از نظر زمان و هزینه بازدارنده است

Increasing Software Reliability

- Fault Avoidance
  -Detect faults statically without executing the program
  -Includes development methodologies, reviews, and program verification
- Fault Detection
  -Detect faults by executing the program
  -Includes testing
- Fault Tolerance
  -Recover from faults at runtime (e.g., transactions)
  -Includes adding redundancy (e.g., n-version programming)

This slide is brought from “Software Architecture and Engineering Testing”, Dr. P. Muller, ETH Zurich: 2018.

What we will talk about next...

- Why Software Testing?