Model-based Student Admission
We should test people the same way we test software.

test levels: Component or Unit testing, Integration testing, Component integration testing, Acceptance testing, System testing

test types: Ad hoc testing, Alpha Testing, All-pairs testing, Beta Testing, Black box testing, Boundary testing, Boundary Value Analysis, Build Verification Test, Code coverage, Compatibility testing, Conformance testing, Combinadic, Exploratory testing, Fuzz testing, GUI software testing, Game testing, Hallway testing, Installation testing, Keyword-driven testing, Load testing, Mobile Device Testing, Monkey test, Manual testing, Model-based testing, Playtest, Pseudolocalization, QuickCheck, Regression testing, Recovery testing, Sanity testing, Scenario testing, Soak testing, Software performance testing, Software verification, Smoke testing, Stress testing, Static testing, Session-based testing, Usability testing, White box testing

Famous bugs: List of software bugs

People: Charles E. Brady, Jr., Kenneth D. Cameron, Patrick G. Forrester, Erich Gamma, Charles D. Gemar, Brent Hailpern, Steven Hawley, Cem Kaner, Adam Kolawa, James D. McCaffrey, Brian Marick, Harlan Mills, Stephen S. Oswald, Gene Spafford

Companies: AutomatedQA, Borland, CTG, Compuware, IBM, Lionbridge, Hewlett Packard HP Software Division, Micro Focus, Microsoft, National Software Testing Laboratories, Segue Software, uTest, Telerik

Test management: Test strategy, Test Plan, Test effort, Test Data Generation

Tools (commercial): AdaTEST95, Automation Anywhere, Cantata++, CAST tool, Coverity, ECLAIR, Goanna, IBM OLIVER (CICS interactive test/debug), Insure++, Jinx, Jtest, LDRA Testbed, HP LoadRunner, HP Quality Center, Microsoft Test Professional, Microsoft Visual Studio Ultimate, QF-Test, Polyspace, Ranorex, Silk Performer, SilkTest, SIMMON, TestComplete, TestPartner, Testware, Time Partition Testing, TOSCA, HP WinRunner, Test Studio

Tools (free/open source): AutoIt, CFUnit, CAMV XML, Check, CPPUnit, Curl-loader, DUnit, Fastest, FindBugs, FitNesse, Framework for Integrated Test, FUnit, HttpUnit, Apache JMeter, JUnit, PHPUnit, Litmus (Mozilla), Mauve (test suite), NUnit, PyUnit, RSpec, Selenium, SimpleTest, soapUI, Splint, STAF, TestNG, Watir, WET Web Test Runner, xUnit

Tools (other) Category: Emulation software, LURCH, Test Automation Framework, Virtual appliance

Certification: British Computer Society, National Software Testing Laboratories, ISTQB, CSTE

Membership associations: Software Engineering Institute, Association for Software Testing, American Society for Quality

Software standards: IEEE 829, TTCN

Terminology: Software bug, Anomaly in software, Test case, Test suite, Test script, Unusual software bug, System under test, Mock object, Test harness, Test data, Testbed, Test bench, Debugger, Boundary case, Verification and Validation, test plan, Test Anything Protocol, Zarro boogs, Thrash (computer science), Memory debugger, Xqa,

Miscellaneous: Software testing outsourcing, Software metric, List of unit testing frameworks

Unit testing

- individual units or modules are tested to determine if they are fit for use
- regular exercises

Crystal Clear, LGPL

https://en.wikipedia.org/wiki/Unit_testing
Stress testing

- determines the robustness of software by testing beyond the limits of normal operation
- comfort zone theory

http://www.dreamstime.com/royalty-free-stock-photos-comfort-zone-image28972918
https://en.wikipedia.org/wiki/Stress_testing_(software)
Compatibility testing

- conducted to evaluate one’s compatibility with the computing environment
- group assignments

https://en.wikipedia.org/wiki/Compatibility_testing
Installation testing

- focuses on what customers will need to do to install and set up the new software successfully

- dry run = internship

https://en.wikipedia.org/wiki/Installation_testing
Case study: UvA MSc SE
UvA MSc SE

- Software Architecture
- Software Specification and Testing
- Requirements Engineering
- Software Evolution
- Software Process
- Software Construction
- Preparation Master Project

http://www.software-engineering-amsterdam.nl/
Student admission @ MSc SE

1. Students
2. Admin check
3. Intake interview
4. Conditionally admitted
5. Premaster
6. Admitted
7. Rejected
Specification-based testing

- The “read the book, talk to me” paradigm
- nonfunctional $\Rightarrow$ hard to test [in a general way]
- Emerging infrastructure
- Is SUT familiar with a notation?
- Can SUT recognise equivalent representations?
Also for flipped classroom
Sets

Relations and functions

Properties of relations

Liskov Substitution Principle

Also for flipped classroom
Prolog encoding

eqv.pl

```
1  eq('≥'(A,B), Q) :- eq('≤'(B,A), Q).
2  eq('≤'(A,B), '∀'(X,'→'(LH,RH))) :- xina(X,A,LH), xina(X,B,RH); xnina(X,B,LH), xnina(X,A,RH).
3  eq('≤'(A,B), '∀'(X,A,E)) :- xina(X,B,E).
4
5  xina(X, A, '∈'(X,A)).
6  % xina(X, A, '¬'(XA)) :- xnina(X,A,XA).
7  xina(X, '∪'(A,B), '∨'(XA,XB)) :- xina(X,A,XA), xina(X,B,XB).
8  xina(X, '∩'(A,B), '∧'(XA,XB)) :- xina(X,A,XA), xina(X,B,XB).
9  xina(X, '¬'(A,B), '¬'(XA,XB)) :- xina(X,A,XA), xnina(X,B,XB).
10  xina(X, '¬'(A), XA) :- xnina(X,A,XA).
11  xnina(X, A, '∉'(X,A)).
12  % xnina(X, A, '¬'(XA)) :- xina(X,A,XA). %,XA /=\ '¬'(\_).
13  xnina(X, '∪'(A,B), '¬'(XA,XB)) :- xnina(X,A,XA), xnina(X,B,XB).
14  xnina(X, '∩'(A,B), '∨'(XA,XB)) :- xnina(X,A,XA), xnina(X,B,XB).
15  xnina(X, '¬'(A,B), '∨'(XA,XB)) :- xnina(X,A,XA), xina(X,B,XB).
16  xnina(X, '¬'(A), XA) :- xina(X,A,XA).
```

https://gist.github.com/grammarware/813374043858030b2059
?- eq('\subseteq'(a,b),'\forall'(x,a,'\in'((x,b)))).
true.

?- eq('\subseteq'(a,b), E).
E = '\forall'(_G276, '\rightarrow'(\in'(\_G276, a), \in'(\_G276, b))) ;
E = '\forall'(_G12, '\rightarrow'('\notin'(\_G12, b), '\notin'(\_G12, a))) ;
E = '\forall'(_G12, a, \in'(_G12, b)) ;
false.

?- eq('\supseteq'(\_G12, a, '\in'(\_G12, b)) ;
false.

?- eq('\supseteq'(\_G12, a, '\in'(\_G12, b)) ;
false.
**Question 7.** Rephrase the following independent statements with relations on sets:

1. \( x \in A \rightarrow x \notin B \)
2. \( x \in A \leftrightarrow x \in B \)
3. \( x \in A \vee x \in B \rightarrow x \in C \)

**Question 8.** Rephrase the following independent statements with quantifiers:

1. \( A \setminus B = \emptyset \)
2. \( 2^A \subseteq 2^B \)
3. \( A \neq \emptyset \)
Question 9. If $c^v_t$ is a substitution of $v$ with $t$ in $c$, then what are the results of the following substitutions?

1. $(\forall x Rxy)^y_z =$
2. $(\forall x \forall y (Rxy \land Ryz))^x_y =$
3. $(\exists y \neg Rxy)^z_y =$
4. $((\exists x Rxy) \land (\forall y Rxy))^x_y =$
**Question 11.** Consider the following grammar:

\[ V ::= a \mid b \mid \cdots \mid y \mid z \]  
\[ C ::= C \bowtie C \mid \pitchfork C \mid C \Join C \mid \triangleright V \triangleright \]

Recognise correct \( C \) terms according to the grammar and draw parse trees of correct ones:

1. \( a \bowtie b \)
2. \( \triangleright p \triangleleft \bowtie \triangleright q \triangleright \)
3. \( \triangleright \pitchfork o \triangleright \)
4. \( \pitchfork \triangleright o \triangleright \)

* cooler = with grammars
Concluding statements

- We know about software testing
- Leverage this for assessment / admission
- Installation testing vs. Acceptance testing
- Exception handling & Destructive testing
- Coverage criteria
- Many more issues to be explored in the future

all photos taken from PESEXLS, CCO Universal license.