ISO/IEC 25010 Software Quality

Functional suitability

Compatibility

Performance efficiency

Usability

Reliability

Security

Maintainability

Portability

Software Requirement

ISO/IEC/IEEE 24765:2017(E)

3.3431

requirement

1. statement that translates or expresses a need and its associated constraints and conditions *[ISO/IEC TS 24748-1:2016 Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management, 2.41; ISO/IEC/IEEE 29148:2011 Systems and software engineering — Life cycle processes — Requirements engineering, 4.1.19]*

2. condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents *[IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes, 3.2]*

3. provision that contains criteria to be fulfilled *[ISO/IEC 14143-2:2011 Information technology — Software measurement — Functional size measurement — Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1, 3.10]*

4. a condition or capability that must be present in a product, service, or result to satisfy a contract or other formally imposed *specification [A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fifth Edition]*

Defect

ISO/IEC 9126
Defect: The nonfulfilment of intended usage requirements
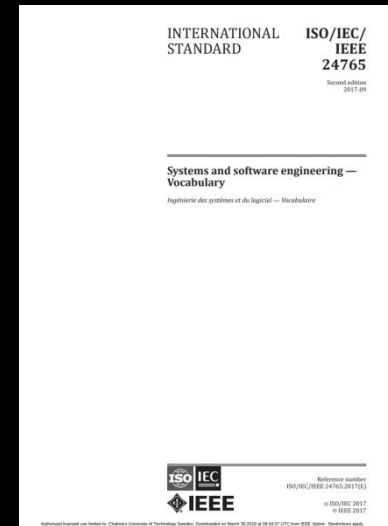**Nonconformity**: The nonfulfilment of specified requirements
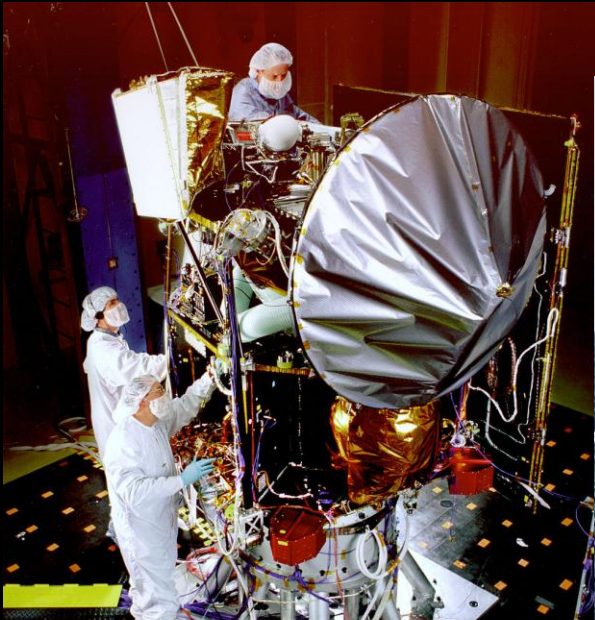
ISO/IEC/IEEE 24765:2010(E)

3.764

defect 1. a problem which, if not corrected, could cause an application to either fail or to produce incorrect results. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual*

2. an imperfection or deficiency in a project component where that component does not meet its requirements or specifications and needs to be either repaired or replaced. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition.*

3. a generic term that can refer to either a fault (cause) or a failure (effect). *IEEE Std 982.1-2005 IEEE Standard Dictionary of Measures of the Software Aspects of Dependability*

The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

‹ Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

| All | New (2 from $1,730,045.91) | Used (15 from $35.54) |

Show ⊙ New ○ ✔Prime offers only (0)

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information
$1,730,045.91 + $3.99 shipping	New	Seller: profnath Seller Rating: ★★★★☆ 93% positive over the past 12 m (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.
$2,198,177.95 + $3.99 shipping	New	Seller: bordeebook Seller Rating: ★★★★☆ 93% positive over the past 12 m (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publishe overstock or have slight shelf wear. Satisfaction guaranteed

Agile & Requirements

Individuals and interactions over

processes and tools

Working software over

comprehensive documentation

Customer collaboration over

contract negotiation

Responding to change over

following a plan

TESTING

Verification

Are we building the software right?

Static

Manual / Automatic inspection

  Loop conditions

  Indices

  Names

  Types

  Argument order

Validation

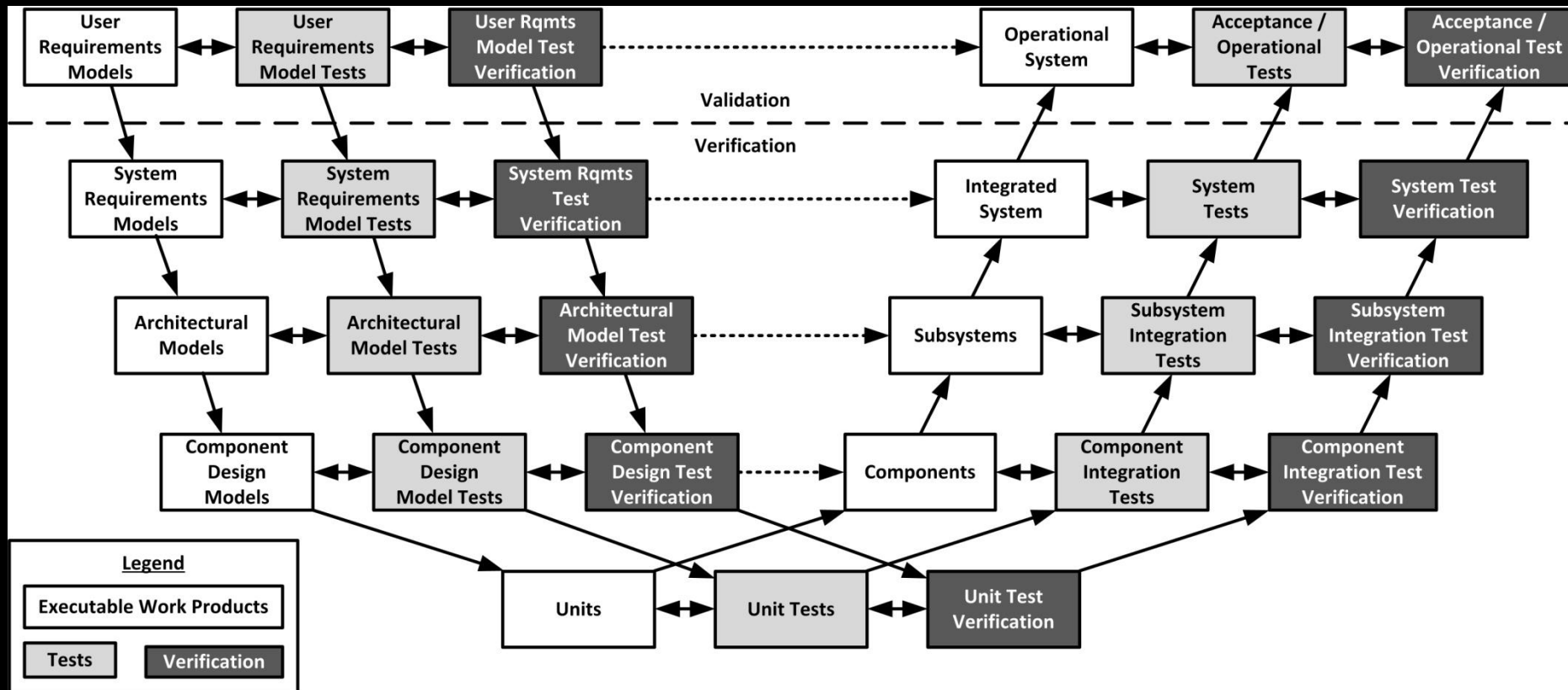Are we building the right software?

Dynamic

Build & Run

Test-to-pass / Test-to-fail

White / Grey / Black

Automatic / Manual

Artefact – Test – Verification

TESTING

Unit testing

System / Integration testing

Acceptance testing

File View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Parms Host port

...ava | RegexTest3BGroupsBackref.java | Test1.java | Test1Test.java | MyAgileTest.java | MyAgileTestTest.java

```java
37          }
38
39  /**
40   * Test of add method, of class MyAgileTest.
41   */
42  @Test
43  public void testAdd() {
44      System.out.println("add");
45      String s1 = "2";
46      String s2 = "2";
47      MyAgileTest instance = new MyAgileTest();
48      int expResult = 4;
49      int result = instance.add(s1, s2);
50      assertEquals(expResult, result);
51  }
52  }
53
```

Files / Services panel:

JavaXpath
Midterm7343
  Source Packages
    <default package>
    agileJunit
      MyAgileTest.java
      Test1.java
    deadlock
    midterm7343
    regex
      RegexTest1Strings.ja
      RegexTest2StringMa
      RegexTest3BGroupsB
      RegexTest3GroupsBa
      RegexTest3PatternM
      RegexTest5Greedy.j
  Test Packages
  Libraries

Navigator panel:

ld - Navigator
s View
MyAgileTestTest
  MyAgileTestTest()
  setUp()
  setUpClass()
  tearDown()
  tearDownClass()
  testAdd()

Test Results / Output / Tasks:

Test Results | Output - Midterm7343 (test) | Tasks

agileJunit.Test1Test x | agileJunit.MyAgileTestTest x
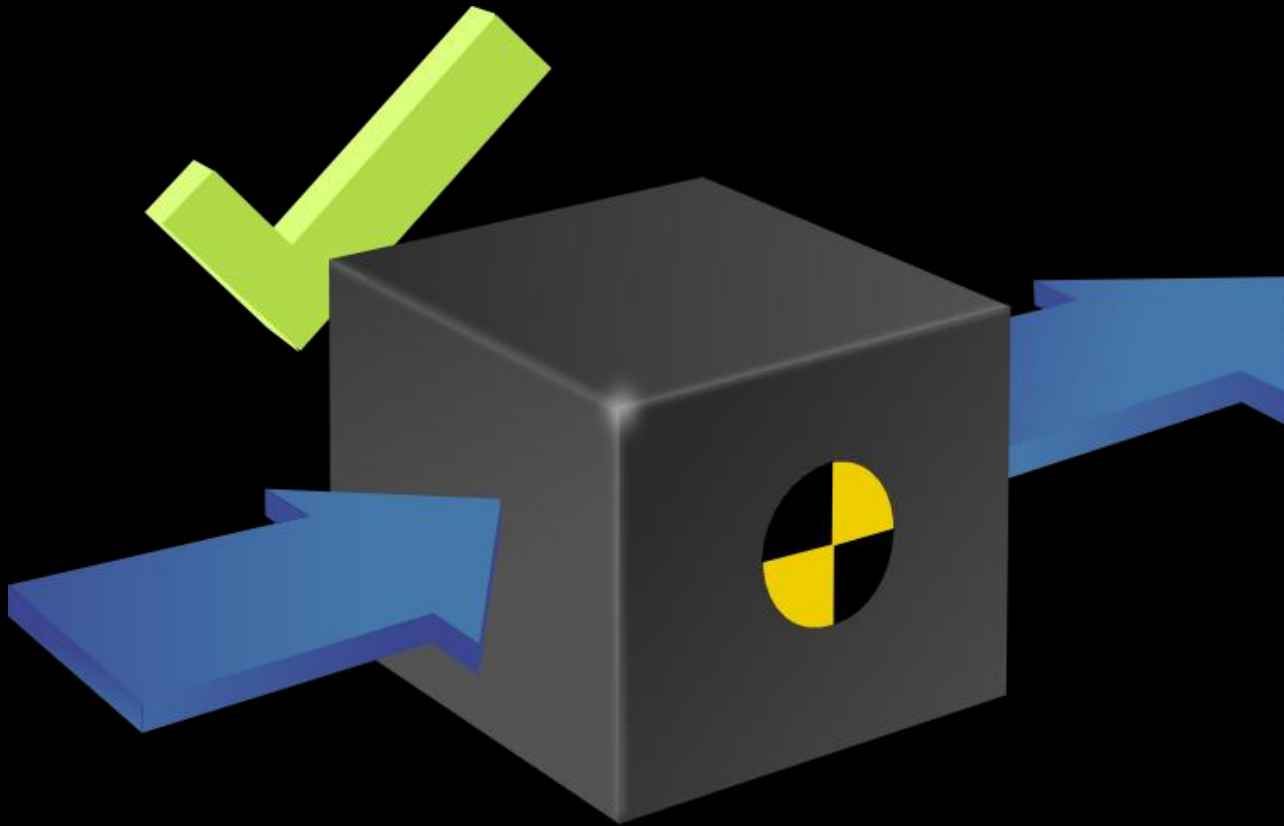
100.00 %

The test passed. (0.059 s)
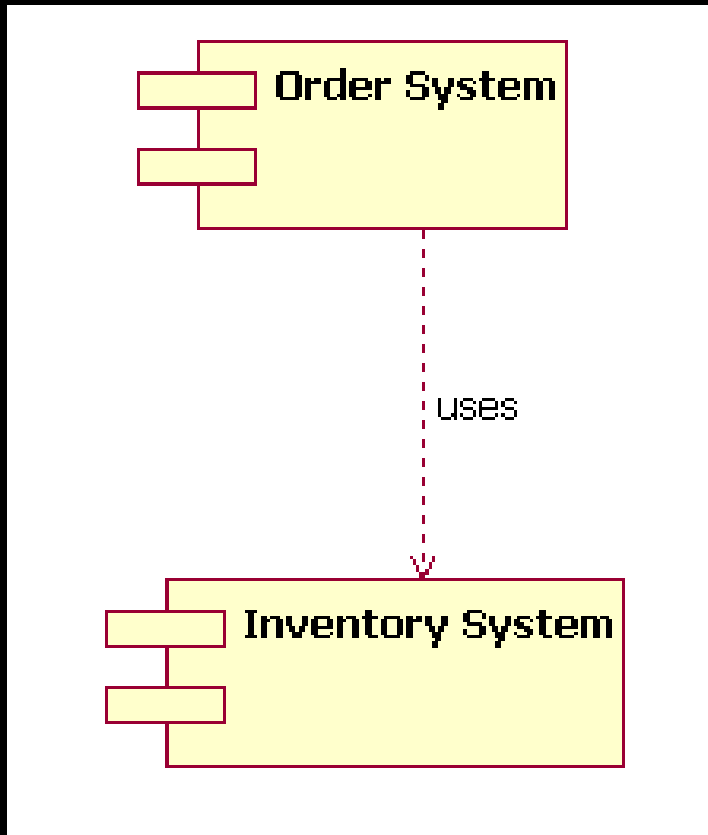 agileJunit.MyAgileTestTest passed

add

System testing

Integration testing

Will Order System and Inventory System behave in an erroneous or unexpected way when working together?
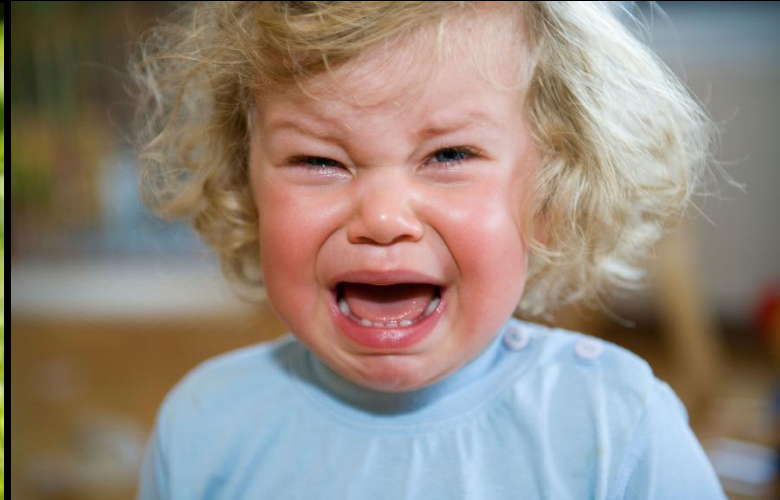
Acceptance Test

USER EXPERIENCE

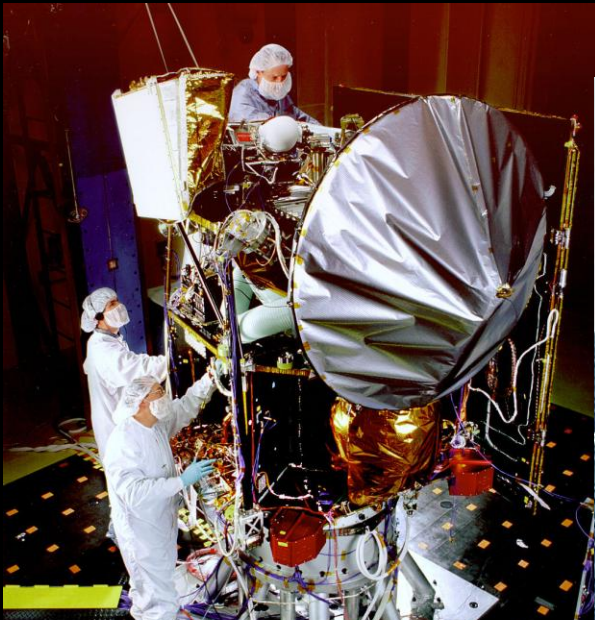A / B TESTS

MULTI-VARIATE

BETA-RELEASE

Strategi för kvalité

Under legoövningen blev våra leveranser inte godtagna efter första sprinten, men det blev de i både andra och tredje. Då kunde vi däremot hela tiden kolla med produktägaren om något vi gjorde höll måtten. Det gjorde det väldigt tydligt om det vi byggde var stabilt och konsekvent.

Så är inte fallet under detta projekt: det kommer inte vara lika tydligt om produkten i projektet är bra jämfört med om ett legobygge är klart och det är inte möjligt att ständigt fråga produktägaren för att undersöka detta.

Som KPI För att mäta vilken kvalitet leveranserna håller tänker vi att mäta hur stor andel av alla rader av vår kod som körs av de enhetstester som vi bygger. Desto större andelen är, desto större är sannolikheten att vi vet vad vår kod gör.

En brist med detta KPI är att det är beroende av våra enhetstester, och det kan finnas fall som vi inte har gjort lämpliga enhetstester för.

The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

‹ Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

| All | New (2 from $1,730,045.91) | Used (15 from $35.54) |

Show ⊙ New ◯ ✔Prime offers only (0)

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information
$1,730,045.91 + $3.99 shipping	New	Seller: profnath Seller Rating: ★★★★☆ 93% positive over the past 12 m (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.
$2,198,177.95 + $3.99 shipping	New	Seller: bordeebook Seller Rating: ★★★★☆ 93% positive over the past 12 m (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publishe overstock or have slight shelf wear. Satisfaction guaranteed

REMEMBER

Specifications will change over time

It's im p o s s ib le to c o m p le te ly te st a p r o g

Testi n gcan t show that a program is bug   -free

A bug can be fixed

Is it a bug or a feature?

Agile & Testing

Acceptance criteria

DoD

Custom ercollaboraton over...

.wfg

Continuous integration

Continuous deployment

Test-driven development

...

TEST-DRIVEN DEVELOPMENT

Add a new test

Run all test cases

Add new code

Run all tests

Refactor

SOME ADVICE

Merge – Merge – Merge

Be aware of your assumptions

Self-organise & take responsibility

Feedback

The customer ≠ The grader

~~Have~~ It's gonna be fun!

PROJECT TRIANGLE

Even More Advice

"It's not the plan that is important,
it is the act of planning"

G. Edwards

"I'm not planning to jump off a bridge with no bun gee"

Norah Jones

REALITY CHECK

What was purpose of lecture?

Which learning objectives were covered? How?

What was the relationship to the course assessment?

42