

Integration and System Testing

Relation to Process Basic & Example Approaches Object-Oriented Issues

ICS 224 5/2/00 1

Integration and System Testing: Main Activities

Requirements Elicitation	Requirements Specification	Architectural Design	Detail Design
<ul style="list-style-type: none"> ✓ Identify qualities ✓ Acceptance test planning 	<ul style="list-style-type: none"> ✓ Validate specifications ✓ System test planning ✓ Create functional tests 	<ul style="list-style-type: none"> ✓ Architectural design inspection ✓ Integration & unit test planning 	<ul style="list-style-type: none"> ✓ Design inspections ✓ Generate oracles ✓ Generate black-box test cases ✓ Automated design analyses

ICS 224 5/2/00 2

Integration and System Testing: Main Activities

Detail Design	Unit Coding	Integration & Delivery	Maintenance
<ul style="list-style-type: none"> ✓ Design inspections ✓ Generate oracles ✓ Unit test planning ✓ Automated design analyses 	<ul style="list-style-type: none"> ✓ Code inspections ✓ Create scaffolding ✓ Unit test execution ✓ Automated code analyses ✓ Coverage analysis 	<ul style="list-style-type: none"> ✓ Integration test execution ✓ System test execution ✓ Acceptance test execution ✓ Deliver regression test suite 	<ul style="list-style-type: none"> ✓ Regression test execution ✓ Revise regression test suite

ICS 224 5/2/00 3

Unit vs. Integration vs. System Testing

Unit Testing	Integration Testing	System Testing
<ul style="list-style-type: none"> from module specifications visibility of code details complex scaffolding required attention to behavior of single modules 	<ul style="list-style-type: none"> from interface specifications visibility of the integration structure some scaffolding required attention to interactions among modules 	<ul style="list-style-type: none"> from requirements specifications no visibility of code no drivers/stubs required attention to system functionalities

ICS 224 5/2/00 4

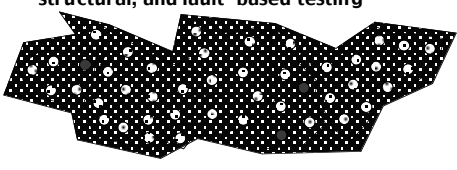
System vs. Acceptance Testing

- **System Testing**
 - the software is compared with the requirements specifications (verification)
 - usually performed by the developer
- **Acceptance Testing**
 - the software is compared with the end-user requirements (validation)
 - usually performed by the customer (buyer)
 - sometime distinguished among α - β -testing for general purpose products

ICS 224 5/2/00 5

Partition Testing

- **Basic idea: Divide program input space into (quasi-) equivalence classes**
 - Underlying idea of specification-based, structural, and fault-based testing



ICS 224 5/2/00 6

Category-Partition Method

Ostrand & Balcer, "The Category-Partition Method for Specifying and Generating Functional Tests." CACM 31(6), June 1988.

STEP 1: Analyze the specification:

- Identify individual functional units that can be tested separately. For each unit identify:
 - parameters and characteristics
 - environment and characteristics
- classify units into categories

STEP 2: Partition the categories into choices

STEP 3: Determine constraints among the choices

STEP 4: Write tests and documentation

ICS 224

5/2/00

7

Category-Partition Method: an example

..... *

Command:

find

Syntax:

find <pattern> <file>

Function:

The find command is used to locate one or more instances of a given pattern in a file. All lines in the file that contain the pattern are written to standard output. A line containing the pattern is written only once, regardless of the number of times the pattern occurs in it.

The pattern is any sequence of characters whose length does not exceed the maximum length of a line in the file. To include a blank in the pattern, the entire pattern must be enclosed in quotes (""). To include a quotation mark in the pattern, two quotes in a row ("") must be used.

* From Ostrand, Balcer, The Category-Partition Method for Specifying and Generating Functional Tests

ICS 224

5/2/00

8

Step 1 – analyze specification: identify categories

- find is an individual function that can be tested separately
- parameters: *pattern, file*
- characteristics (pattern)
 - explicit (immediately derivable from specs):
 - pattern length
 - pattern enclosed in quotes
 - pattern contains blanks
 - pattern contains enclosed quotes
 - implicit ("hidden" in specs):
 - quoted patterns with/without blanks
 - several successive quotes included in the pattern
 -

ICS 224

5/2/00

9

Step 2 – partition categories

Parameters:

Pattern size:

empty
single character
many characters
longer than any line in the file

Quoting:

pattern is quoted
pattern is not quoted
pattern is improperly quoted

Embedded blanks:

no
one
several

Parameters (cont...)

Embedded quotes:

no
one
several

File name:

.....

Environment:

Number of occurrences of pattern in a file:

none
one
several

Pattern occurrences on target line:

.....

ICS 224

5/2/00

10

Step 3 – determine constraints

Parameters:

Pattern size:

empty [property Empty]
single character [property NonEmpty]
many characters [property NonEmpty]
longer than any line in the file [error]

Quoting:

pattern is quotes [property Quoted]
pattern is not quotes [if NonEmpty]
pattern is improperly quotes [error]

.....

Environment:

Number of occurrence of pattern in a file:

none [if NonEmpty] [Single]
one [if NonEmpty] [property Match]
Match

ICS 224

5/2/00

11

Some Considerations on the Category Partition Method

- a practical implementation of general principles:
 - partition testing
 - boundary testing
 - erroneous conditions
- other approaches with similar goals, but different procedures:
 - condition tables
 - cause effect graphs
 - equivalence partitioning

ICS 224

5/2/00

12

Exercise

from the EXCEL manual:

“AutoSum Button

Automatically invokes the SUM function and suggests the range of cells

to be added. To use the AutoSum button, select a cell adjacent to a row or column of numbers you want to add, and click the AutoSum button on the Standard toolbar.

If Microsoft Excel suggests an incorrect range, drag through the range you want, and press ENTER to accept the completed formula.”

- **identify and partition categories according to the Category Partition Method**

ICS 224

5/2/00

13

A possible solution

categories (and characteristics)

- **selected cell (parameter)**
 - position
- **range selection mode (parameter)**
- **selected range (parameter)**
 - shape
 - size
 - position relative to the selected cell
- **spreadsheet (environment)**
 - size
 - type of cells
 - value of cells

ICS 224

5/2/00

14

A possible solution (cont.)

partition categories

- **position of the selected cell**
 - first cell (a 1)
 - first row
 - first column
 - not on the border
- **range selection mode**
 - automatically
 - manually
- **shape of the selected range**
 - set of consecutive cells in a single row
 - set of consecutive cells in a single column
 - rectangular set of cells
 - non-adjacent sets of cells

ICS 224

5/2/00

15

A possible solution (cont.)

- **size of the selected range**
 - empty
 - one cell
 - many cells
- **position of the selected range**
 - adjacent to the selected cell
 - non-adjacent to the selected cell
 - overlapping the selected cell
-

ICS 224

5/2/00

16

Integration testing

- **Testing of components and subsystems during integration**
- **Integration of well tested (correct) modules may reveal defects due to:**
 - Bad use of the interfaces (bad interface specifications | implementation)
 - Wrong hypothesis on the behavior of related modules (bad functional specification | implementation)
 - Use of poor drivers/stubs: a module may behave correctly with (simple) drivers/stubs, but result in failures when integrated with actual (complex) modules.

ICS 224

5/2/00

17

Testing during Design

- **Primary focus of testing during architectural design and module design should be integration test planning**
 - **Focus in design is decomposition**
 - architectural design defines *components, connectors, and their configuration*
 - module design defines *modules, module interfaces, and their interactions*
 - **Focus in integration is composition**
 - **Focus in integration testing is verifying and/or validating the interactions between components and modules**

ICS 224

5/2/00

18

