

[Presentation](#)

[Bio](#)

[Return to Main Menu](#)

P R E S E N T A T I O N

TG1

Thursday, Dec 7, 2000

The Three Holy Grails of Test Development

Hans Buwalda



The 3 Holy Grails of Test Development



Hans Buwalda
CMG - TestFrame Research Centre
hans.buwalda@cmg.nl
www.testframe.com

The Quest

The Challenges for a Test Process

- Testing should be fun
- Testing should be effective
- Testing should be efficient
- Testing should be under control

Common problems with testing

- Time consuming
- Costly
- Tends to be neglected
- Experienced as boring to do (in particular the execution)
- Hard to start in time

Even more common problems with testing

- Difficult to manage:
 - What is the progress
 - What is the quality
- The proper resources (users, specialists) are not (made) available when needed
- Lack of test design competency
- etcetera

Dimensions of Testing

- Organisation and management
- Test development
- Test automation

Re-usable test products

| A | B | C | D |
|----------------------|---------|-------|-----|
| ... | | | |
| <i>transfer</i> | Houston | Klein | 210 |
| <i>check balance</i> | Klein | 210 | |
| <i>transfer</i> | Savy | Klein | 150 |
| <i>check balance</i> | Klein | 360 | |
| ... | | | |

test cluster

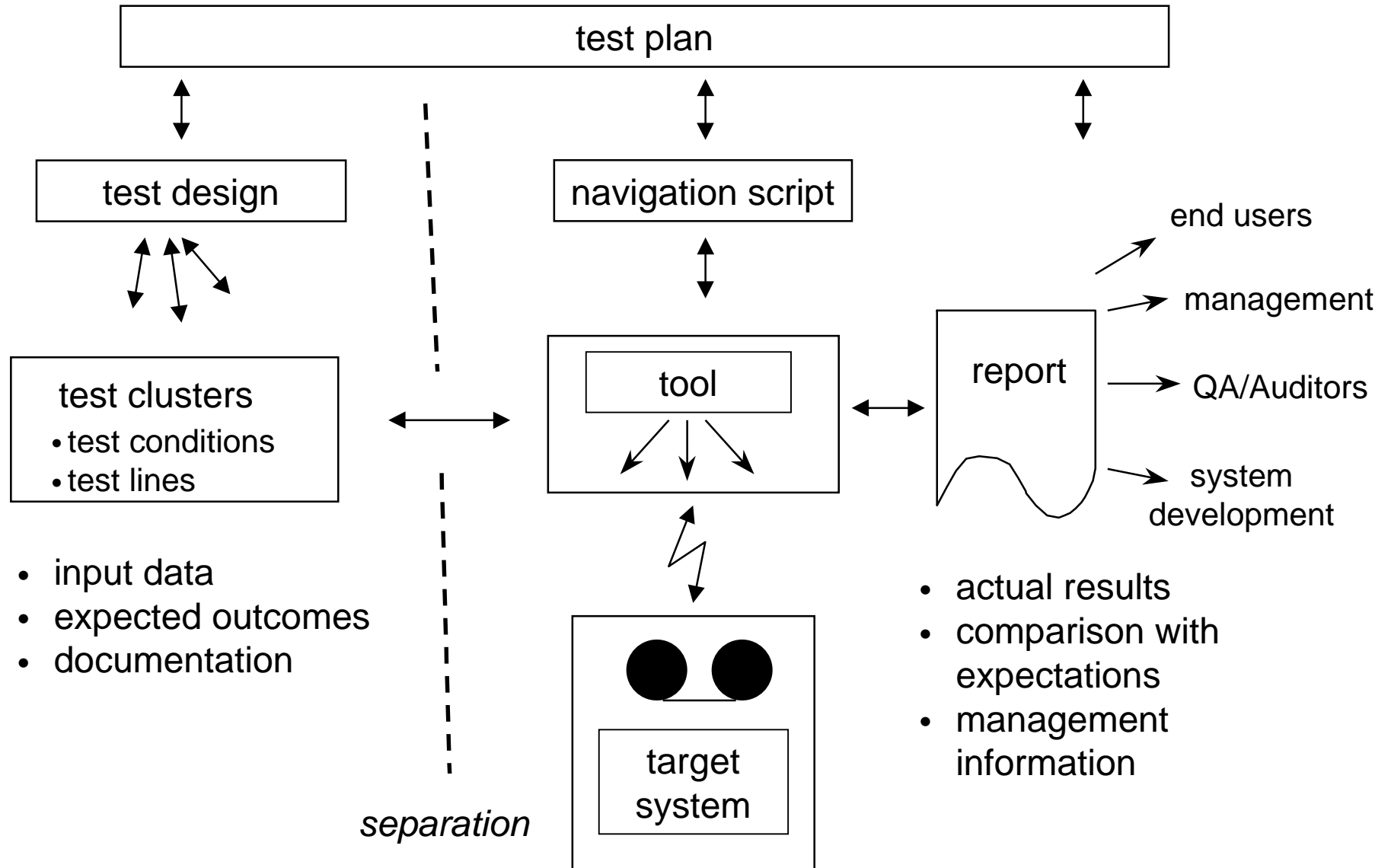
test development

navigation scheme

test execution

| | |
|-----------------------|--|
| ... | |
| check balance | |
| enter customer | |
| ... | |

The product life cycle

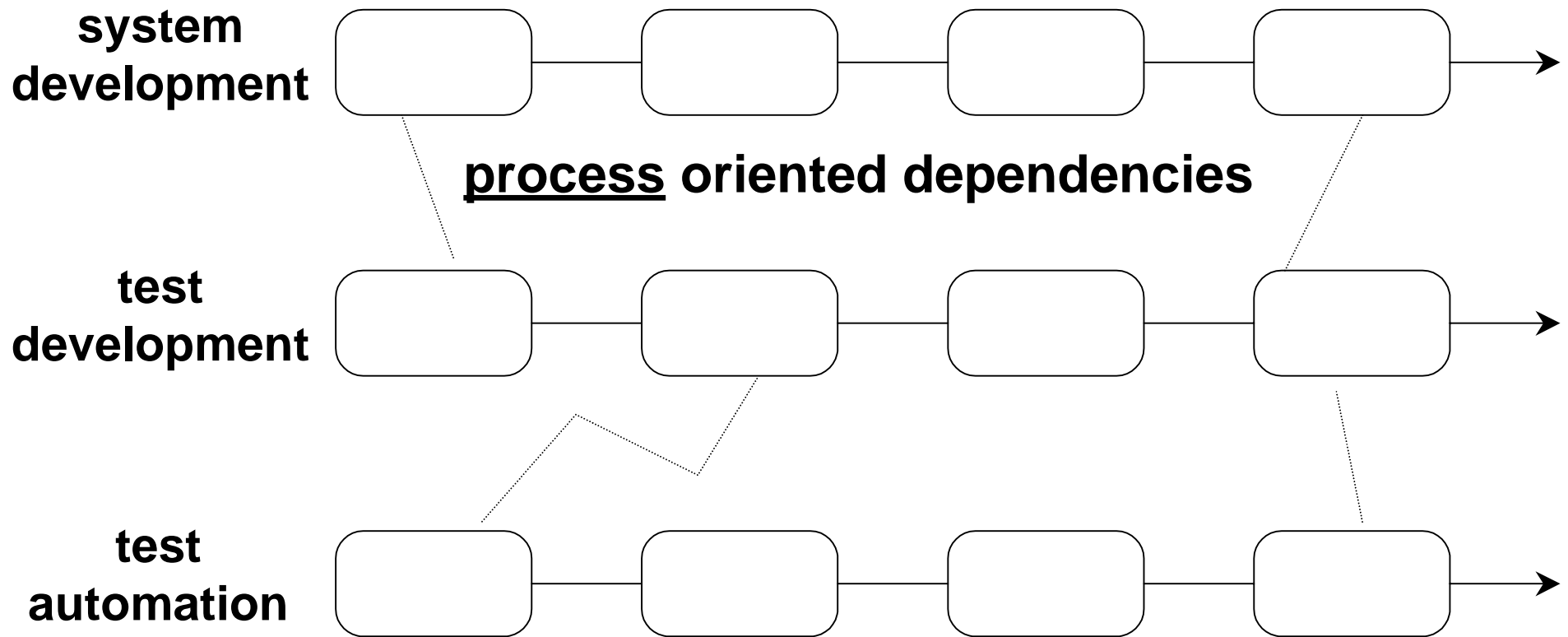


Example Test Condition

| nr | description |
|------|--|
| ... | |
| 3.51 | it is checked that the exit date is after the entry date |
| ... | |

| <i>test condition</i> | 3.51 | name | entry date | exit date |
|----------------------------|------|---|------------|------------|
| <i>enter employment</i> | | Bill Goodfellow | 1999-10-02 | 1999-10-01 |
| <i>check error message</i> | | The exit date must be after the entry date. | | |

Independence of life cycles



The Holy Grail

- Part of the legend of king arthur
- Most of all a symbol
- To find the holy grail one has to be the perfect knight
- To search for it is as important as finding it

The Holy Grail as a model

- General principles, good to “strive” for, but difficult to fully achieve
- The three principles suggested in this presentation are for test development
- Coming close to the grails will help quite a bit
- Being far away is a good recipe for trouble
- The first “victim” is the manageability

The three “holy grails” of Test Development



effective “clustering” of tests



choosing the right approach per cluster



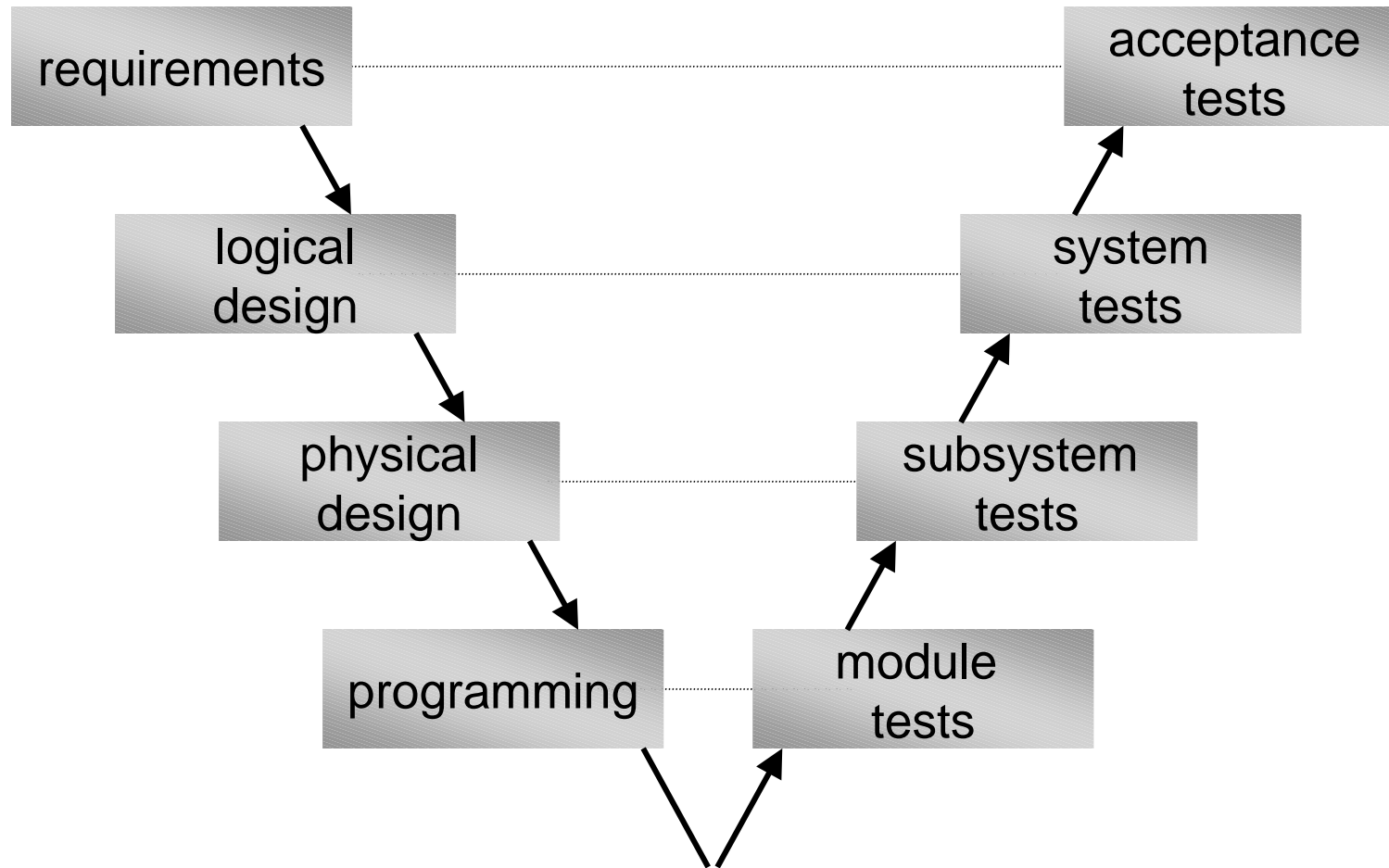
the proper level of actions



The First Grail

clustering

well known, the V Model (simplified)



(inflexibel) use of the V Model can involve risks

(see Brian Maricks article: www.testing.com/writings/new-models.pdf)

The “cluster model” as an alternative

- fit to project situation as an explicit step
- V Model is “member of the class”
- further decisions are per cluster:
 - how -> which technique, automate or not
 - when -> develop, execute
 - who -> stake holders, tester, auditor, ...
 - why -> business risk, complexity, *made by John*, ...

What is not likely to be good clustering

- combinations of low and high level tests
- when you have to change all of them for a new system release
- if all clusters look alike
- clusters are dependent on each other
- you can't start developing them now

Typical consequences / symptoms

- complaining people
- unnecessary high test maintenance (high impact of system changes on the test)
- difficulties in running any test
- “unpleasant” test process
- no (sense of) control

Clustering recommendations

- Logical to all concerned
- Independent from other clusters
- Well differentiated and clear in scope
- Fitting the priorities and planning of the project
- Balanced in size and amount

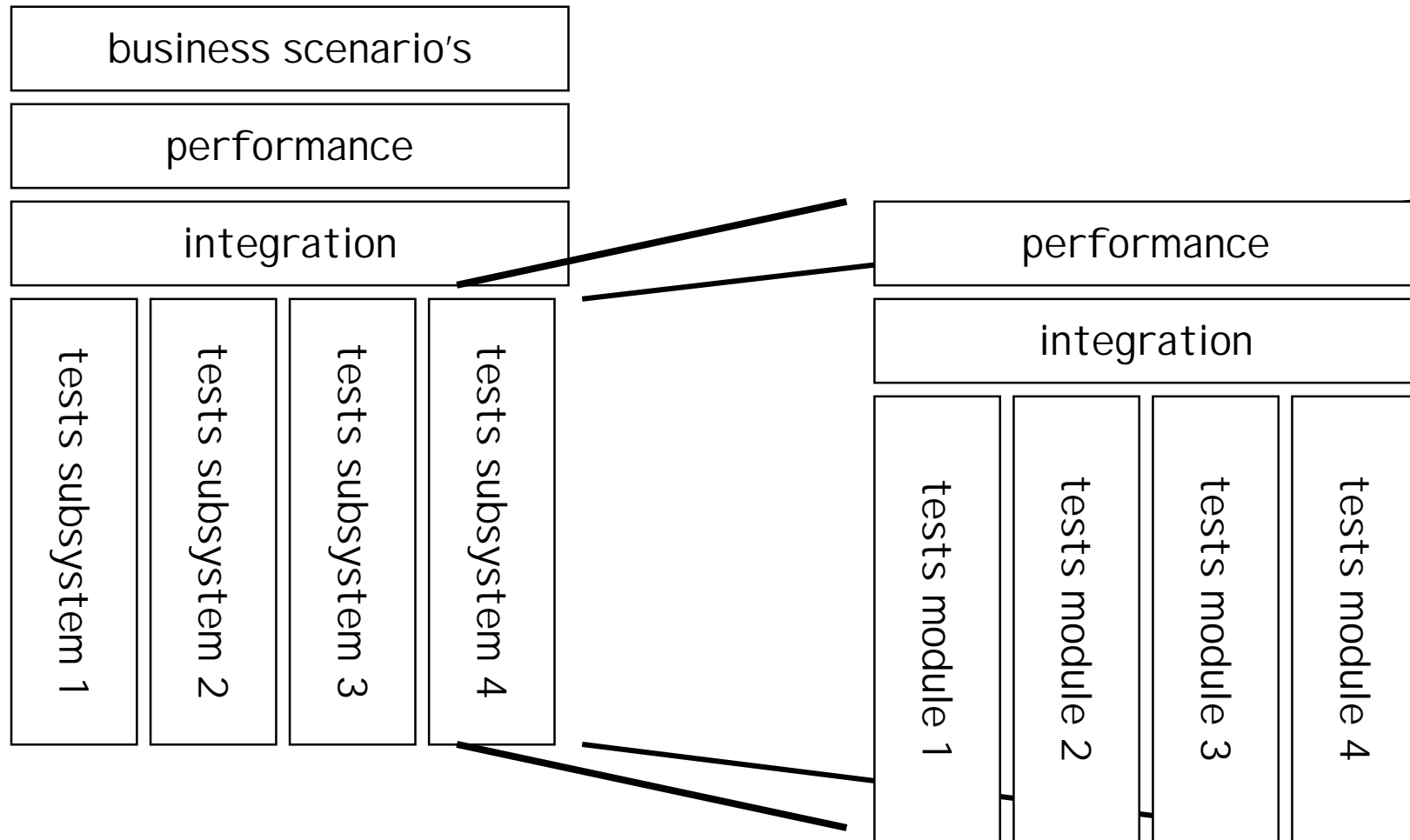
Some angles for clustering

- business and failure risks
- planning and control
- test execution
- complexity of the test
- stake holders and others involved
- part of the system
- detail level
- quality aspect
- technique used for the test (double role)
- functional and technical requirements
- . . .

Example of a (customer friendly) business risks analysis at a financial institution

| | | | |
|--|--|-----------------------------|----------------------------|
| <p><i>if this goes wrong then...</i></p> | <p><i>... there are financial (or worse) consequences for our customer</i></p> | <p><i>all customers</i></p> | <p>must have</p> |
| | | <p><i>one customer</i></p> | <p>should have</p> |
| | <p><i>... there are non-financial consequences for our customer</i></p> | <p><i>all customers</i></p> | <p>should have</p> |
| | | <p><i>one customer</i></p> | <p>could have</p> |
| | <p><i>... there are financial (or worse) consequences for ourselves</i></p> | <p><i>no workaround</i></p> | <p>should have</p> |
| | | <p><i>workaround</i></p> | <p>nice to have</p> |
| | <p><i>... there are non-financial consequences for ourselves</i></p> | <p><i>no workaround</i></p> | <p>should have</p> |
| | | <p><i>workaround</i></p> | <p>nice to have</p> |

The clustering process (example)





The Second Grail

approach per cluster

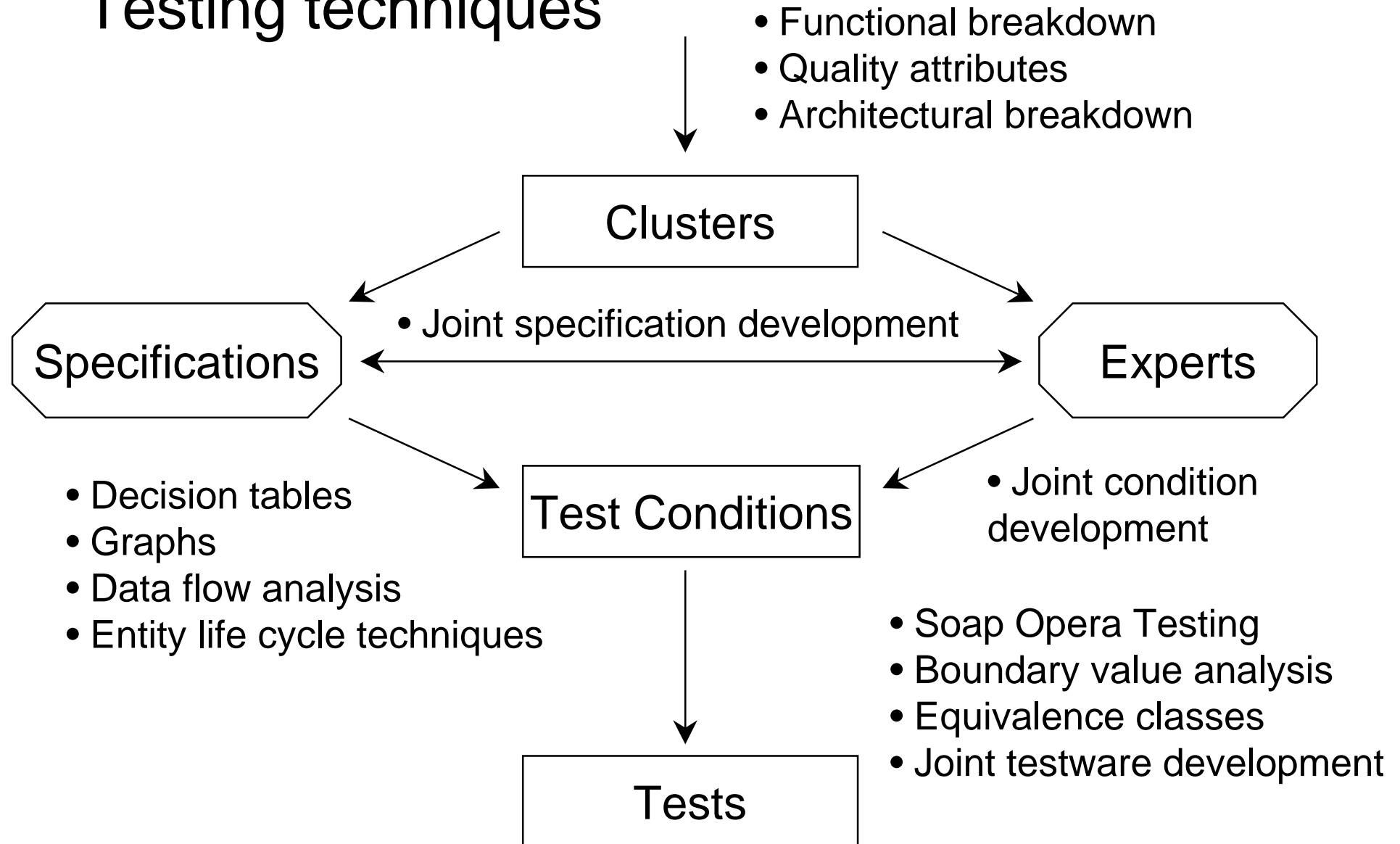
What is not a good approach (..)

- Complaining people
- The wrong level of techniques, like using a boundary value analysis in a high level business test
- Lots of hard labour without interesting results (“monks work”)

Common Design Techniques Summary

- Equivalence Partitioning
- Boundary Value Analysis
- Output Forcing
- State Models
- Error Guessing
- Reference: [Beizer, 1990] [Kit, 1996]

Testing techniques



Soap Operas



Ashley hears about Jack's deposit when he thought he had to go. Victoria lectures her father about what's wrong with him and Nikki but Victor advises her that it's none of her business. Olivia learns Dru has no regrets about leaving and takes great satisfaction in having Lily as her companion. Dru then asks Olivia why she is raking Malcolm over the coals. Stopping by Gina's, Nikki spots Brad and sits with him, admitting she doesn't want to be alone tonight. Victor stops by Mack's party at the Crimson Lights. Ashley takes a home pregnancy test. Worried about Billy, Raul makes call and J.T. claims he doesn't know where Billy is. Raul rushes over and finds Billy out cold in the snow. Raul worries when he can't find a pulse. . . .

Properties of Soap Operas

- About “real life”
- But condensed
- And more extreme

Soap Operas for testing

- Define a scope of the test to develop
- Identify with the business environment
- Which elements would make things difficult
- Draft scenario's (typical some dozen lines)
- Write them down in clusters

Examples of story lines when used for testing

Pension Fund

William starts as a metal worker for Industrial Entropy Incorporated in 1955. During his career he becomes ill, works part time, marries, divorces, marries again, gets 3 children, one of which dies, then his wife dies and he marries again and gets 2 more children....

World Wide Transaction System for an international Bank

A fish trade company in Japan makes a payment to a vendor on Iceland. It should have been a payment in Icelandic Kronur, but it was done in Yen instead. The error is discovered after 9 days and the payment is revised and corrected, however, the interest calculation (value dating)...

Example of test lines

| | from | to | amount | valuta | trans nr |
|--------------------|-----------|------------|------------|--------|-----------|
| enter payment | 123421344 | 4124244123 | 120000 | yen | &keep tx1 |
| check value dating | &tx1 | \$0.47 | | | |
| wait days | 9 | | | | |
| | | | | | |
| order to reverse | &tx1 | | | | |
| | | | | | |
| | from | to | amount | valuta | trans nr |
| enter payment | 123421344 | 4124244123 | 1200000000 | IKr | &keep tx2 |
| check value dating | &tx2 | \$7,701.56 | | | |
| | | | | | |
| | | | | | |

Soap Operas (in testing) are not necessarily:

- “Extreme”
- Far fetched
- Long and elaborate
- Pieces of art and creativity

“Killer Soaps”

- More specifically aimed at finding hidden problems
- Run when everything else has passed
- One option: put a killer soap at the end of a normal cluster
- Ask the “specialists” for input

Test Design Templates

- Spreadsheet based technique for designing tests
- Introduced in 1998 as an extension to testframe by edward kit
- Friendly technique and yet very strong, for example in identifying needed situations and combinations
- For a further description please have a look at ed's article in software development magazine:
<http://www.sdmagazine.com/breakrm/features/s992f2.shtml>

Example of Test Design Templates

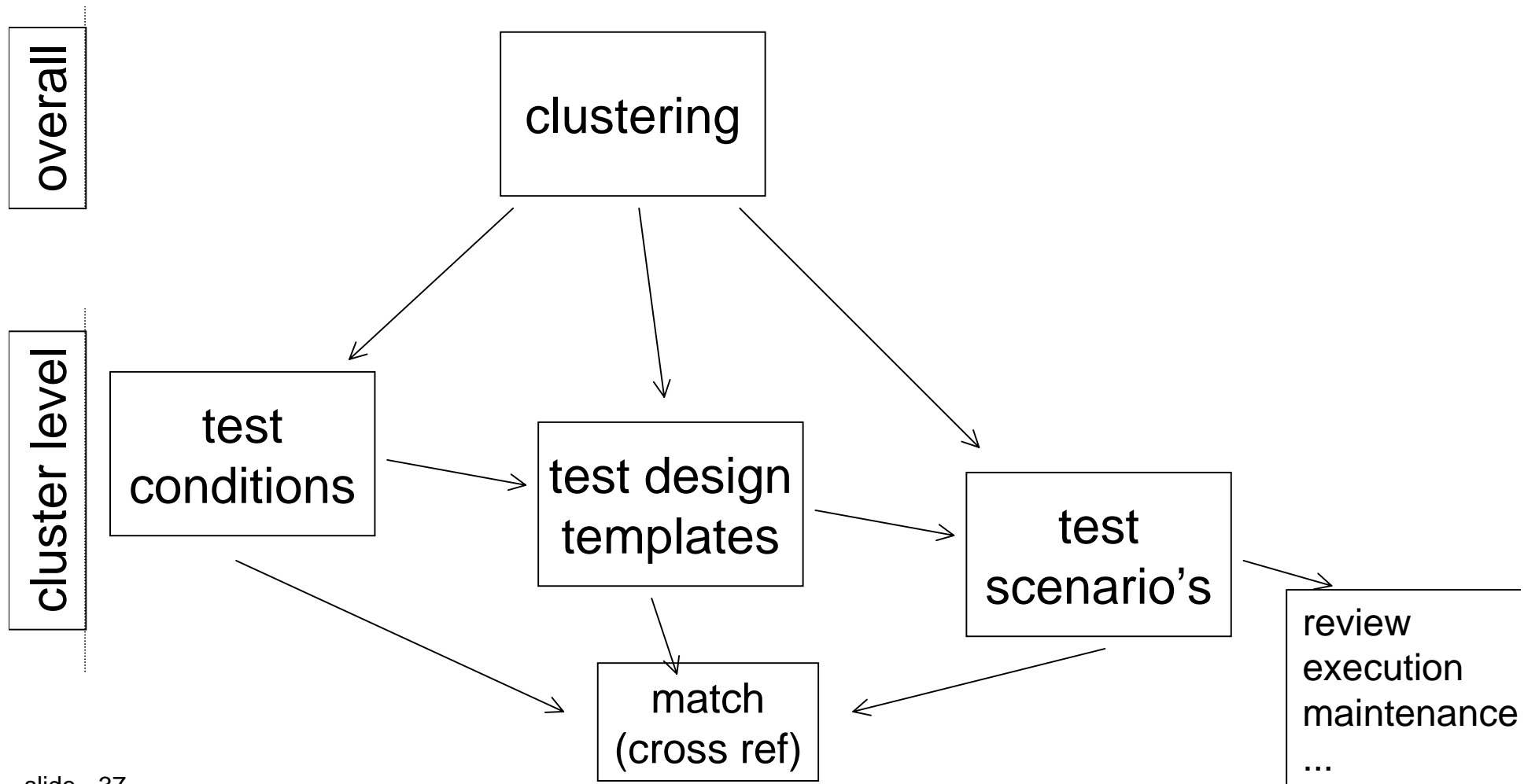
(applied to SDT's ReviewPro package for supporting inspections and reviews)

| | | | | | |
|----------------------------|--|--|---|--|--------------------------------------|
| Feature Hierarchy: | Default Entry Type Standard | | | | |
| Matrix Summary: | Check that the default (non-customized) Entry Types can be selected/used | | | | |
| Test Design: | Technique | Equivalence Class <input type="button" value="v"/> | | Feature Combination | No <input type="button" value="v"/> |
| Risk Analysis: | Impact | High <input type="button" value="v"/> | | Likelihood | Low <input type="button" value="v"/> |
| Test Case ID: | DETS01 | DETS02 | DETS03 | DETS04 | |
| Test Case Validity: | Valid <input type="button" value="v"/> | Valid <input type="button" value="v"/> | Valid <input type="button" value="v"/> | Valid <input type="button" value="v"/> | <input type="button" value="v"/> |
| Priority: | High <input type="button" value="v"/> | High <input type="button" value="v"/> | High <input type="button" value="v"/> | High <input type="button" value="v"/> | <input type="button" value="v"/> |
| Test Condition | | | | | |
| Log Entry Input Fields | * | * | * | * | |
| Document Reviewed = | | | | | |
| Location = | | | | | |
| Type = | defect | question | external issue | praise | |
| Severity = | | | | | |
| Status = | | | | | |
| Disposition Code = | | | | | |
| Summary = | | | | | |
| Attachments = | | | | | |
| Select Default Entry | * | | | | |
| Select Entry #2 | | * | | | |
| Select Entry #3 | | | * | | |
| Select Entry #4 | | | | * | |
| Expected Results | Output= Go to Reviewer's Log View, Log is in the Reviewer's Log Forum, Correct Entry Type is selected in the Log Entry | Output= Go to Reviewer's Log View, Log is in the Reviewer's Log Forum, Correct Entry Type is selected in the Log Entry | Output= Go to Reviewer's Log View, Log is in the Reviewer's Log Forum, Correct Entry Type is selected in the Log Entry | Output= Go to Reviewer's Log View, Log is in the Reviewer's Log Forum, Correct Entry Type is selected in the Log Entry | |

Usage of Test Design Templates as portal to Soap Operas

| Template ID: | MB des 1 | MB des 2 | MB des 3 | MB des 4 |
|---------------------|----------|----------|------------|----------|
| | | | | |
| customer | * | * | * | * |
| last name | | | | |
| first name | | | | |
| balance | positive | too low | positive | positive |
| number | | | | |
| confirmation letter | yes | yes | | |
| automatic numbering | | | | yes |
| | | | | |
| matching → | | | | |
| tested in scenario: | MB01 | | MB01, MB02 | MB02 |

A life cycle for test development



What can joint sessions give you

- Test Strategy
- Acceptance Criteria
- Cluster Grouping
- Test conditions
- Evaluation of Results
- Starting up development of scenarios

Joint Testware Development (JTD)TM

- Moderator / chairman
- Users
- Business specialists
- Developers
- Testers

Set-up of a joint session for a telecom provider

- **1st session**
 - Introduction by moderator and project manager
 - explanation about the JTD procedure
 - explanation of the functional area by a specialised user
- **2nd session**
 - start of production of test conditions
- **3rd session**
 - start of production of test scenarios
- **4th session**
 - evaluation test scenarios



The Third Grail

level of actions

Composition of Action Words

- Specification of an action, a check or a documentary statement
- Communication between Navigation and Test Cases
- Consistent
- Standard
- By-product of the test analysis

What is probably not a good action level

- “test all”
- low level “navigation” in a high level test:

```
type           "Hans"  
press key     <tab>  
type          "Buwalda"  
click button  OK  
...
```

- unclear “insiders only” language

```
set code      Fc122x      XX33  
...
```

Typical consequences / symptoms

- tests become quite unreadable (especially for non experts)
- unpleasant work to make the tests
- hard to understand the results
- high risk of mistakes
- and, of course, heavy maintenance dependency

Designing Action Words

- What actions does the test tool perform with a specific Action Word
- Scope of the test determines the Action Word level
- Manage the set of Action Words
- Document information about the Action Words

Low-level

- Low-level functions take care of the communication with the application
- Low-level functions are being called from high level action word functions
- Use of low-level functions enhances maintainability
- Typical examples:
 - push button
 - select list box item

Intermediate level

- Optional, use for complex navigation's
- Navigation at level of:
 - Window
 - Record
 - Message
 - ...
- Called as subroutine by high level action words

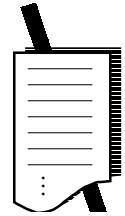
High level

- Defined by testers,
 - (not by navigators and not in advance)
- Specification of the test, navigation:
 - Adds default values
 - Moves across windows
 - Takes care of unexpected situations
 - ...
- Uses low level and intermediate level

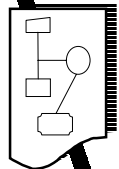
system development

The life cycle

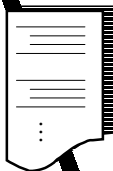
test development



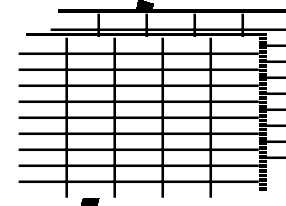
specifications



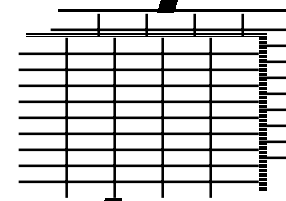
design



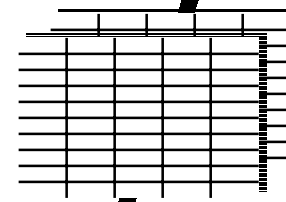
programming



- high level business oriented tests
- production acceptance tests



- functional tests
- technical tests



- low level functional tests
- technical tests

navigation

high level actions

intermediate level actions

low level actions

Acknowledgements

- Edward Kit
- Dorothy Graham
- Brian Marick
- Dirk van Dael
- Erik Jansen
- Dennis Janssen

References

- Beizer, Boris, *Software Testing Techniques, Second Edition*, Van Nostrand Reinhold, 1990
- Buwalda, Hans, *Testing with Action Words*, STAR May 1998
- Buwalda, Hans, and Kasdorp, Maartje, *Getting Automated Testing Under Control*, STQE Magazine, November 1999
- Buwalda, Hans, *Soap Opera Testing*, STAR East May 2000
- Graham, Dorothy, and Fewster, Mark, *Automating Software Testing*, 1999
- Kit, Edward, *Software Testing in the Real World*, Addison Wesley Longman, 1996
- Myers, Glenford, *The Art Of Software Testing*, Wiley Interscience, 1979
- Kit, Edward, *Integrated, Effective Test Design and Automation*, Software Development Magazine, February 1999

Thursday 7 December 2000

Keynote 1

The Three Holy Grails of Test Development

Hans Buwalda

Hans Buwalda is project director at CMG, a leading European information technology services group. He is responsible for new developments around the TestFrame approach for testing and test automation of which he is the main architect. In 1996 he presented the main ideas for the first time to an international audience in a speech called 'Testing with Action Words, abandoning record and playback'. Since then the method is being used in an increasing number of countries and Hans has become a frequent speaker at industry conference, tutorials and workshops.

