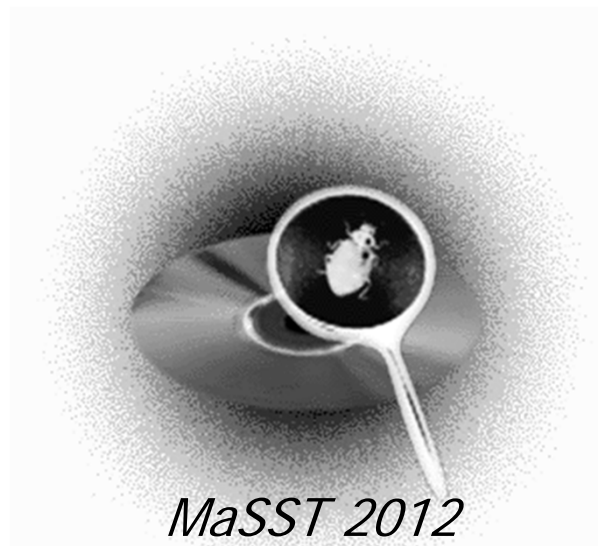


# *Viewpoint-based Test Architecture Design*



*MaSST 2012*

*(International workshop on Metrics and Standards for Software Testing)  
in conjunction with SERE 2012 Washington D.C.*

NISHI, Yasuharu  
The university of Electro-Communications, Tokyo

© NISHI, Yasuharu

# Profile – NISHI, Yasuharu

---

Assistant professor:

the University of Electro-Communications, Japan

(also providing consultancy service to industry on testing and TQM)

President:

Association of Software Test Engineering, Japan (ASTER)

President:

Japan Software Testing Qualifications Board (JSTQB)

National delegate:

ISO/IEC JTC1/SC7/WG26 Software testing

Founder:

Japan Symposium on Software Testing (JaSST)

Founder:

Testing Engineers' Forum (Japanese community on software testing)

Vice chair:

SQiP/Software Quality Committee of JUSE (promoting organization of TQM)

(SQiP has published the book of "SQuBOK: Software Quality Body of Knowledge" and is operating engineer certification on software quality)



# Outline of presentation

---

- Research area on test architecture is necessary
- I propose Viewpoint-based Test Architecture Design
  - Viewpoints / Relationships / Containers / Notation: NGT / Process model: VSTeP / Patterns / Refinement / Styles / Quality attributes of test suite
- Viewpoint-based Test Architecture Design could contribute several standards
  - ISO/IEC DIS 29119s / IEEE std. 829-2008 / ISO/IEC 25000s / ISO/IEC 12207 ?
- Viewpoint-based Test Architecture Design can make several research area such as:
  - Test architecture patterns
  - Test architecture styles
  - Quality characteristics of test suite
  - Detail test architecture design process
  - Better notations
  - Architecture-level reuse / product lines of test suite
  - Metrics for test architecture ?

# Research area on test architecture is necessary

---

- Why did software architecture get necessary ?
  - Software got huge and complicated since 1990s
  - Software escaped from just bundle of hardware since 1980s
- Now test architecture also gets necessary !
  - Test cases (= test suite) also get huge and complicated such as
    - » a test project with over 100,000 test cases
    - » over 10 test levels
    - » various test types such as load, configuration and security
  - Test suite shouldn't be just bundle of test bases, e.g. requirements
    - » Test suite should sometimes be designed independently from requirements because software requirement bugs gets so increased as requirement gets huge and complicated

test suite itself is "artifact"

# Test suite is “huge artifact”, NOT just checklist

---



Huge artifact has  
“architecture”

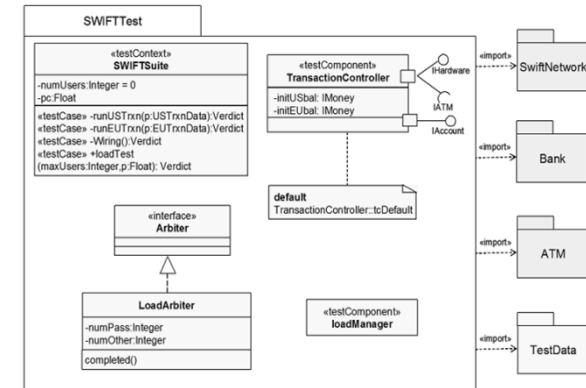
# No research area on Test Architecture Design

---

- Software architecture consists of modules and relationships
  - Software design has two layers: architecture design and detail design
    - » Architecture design focuses on design of modules and relationships among them
    - » Detail design focuses on design of a module inside
- Test design should also have two layers:
  - Test architecture design:
    - » Design of test levels and test types, and relationships among them
  - Test detail design:
    - » Design of a test level or a test type inside, e.g. path testing
- There is no research area on test architecture design which corresponds to software architecture design
  - Nowadays “Test design” means test detail design, i.e. just how to derive test cases after selecting test levels or types
  - Test levels or types can correspond to “module” of software

# Test “system” architecture and test “suite” architecture

- U2TP defines ‘test architecture’ as test “system” architecture
  - U2TP: UML 2.0 Test Profile
  - Architecture for software development has two types as software architecture and system architecture
    - » Software architecture focuses on software inside
    - » System architecture focuses on execution environment
  - The concept of ‘test architecture’ of U2TP focuses not on architecture of test suite but on execution environment including automation
    - » In other words,  
U2TP mainly focuses on Test “system” architecture but we should also research on Test “suite” architecture
  - The concept of Test Architecture in this presentation is test “suite” architecture



# Lack of Test Architecture Design research

---

- Comparison of development process and test process also shows lack of research on test architecture design



*Typical software development process*



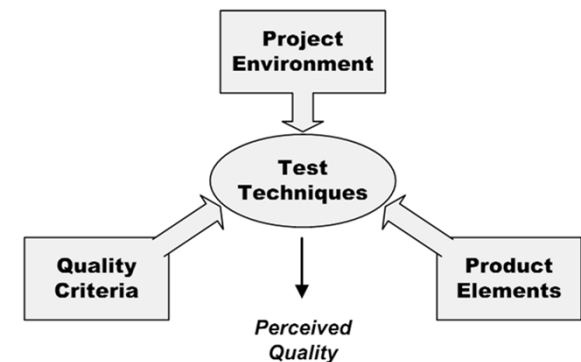
*Part of typical test process*



# Does “Planning” sound design work ?

---

- “Planning” sounds a management word
  - PMBOK Guide defines Planning as one of basic process groups for project management
  - Software test experts traditionally recognize part of test planning as test architecture design work because they have to reflect rapidly management issue to test cases
  - Test strategy may be similar word to test planning
- We should use a design word for design work
  - Test planning or test strategy actually includes design of overall test suite like test architecture
  - But we should use the word “Test Architecture Design” for design work because management issues will be far more emphasized in using the words “test planning” or “test strategy” than engineering issues such as models and patterns
    - » I don’t deny using the word “test planning” if distinguished from design work



J. Bach, “Heuristic Test Strategy Model”  
© NISHI, Yasuharu

# Technologies for software/test architecture

---

## For software architecture

- Abstraction
  - Ex) Object oriented paradigm
- Separation of concerns
  - Ex) Encapsulation
- Relationship with NFR
  - Ex) Quality characteristics of software
- Modeling techniques / Patterns
  - Ex) Design patterns (MVC et al.)
- Notations / meta-modeling
  - Ex) UML  
Arch. Description Language

## For test architecture

- Abstraction
  - Creating integrated concept of high level test case, test type and test level
- Separation of concerns
  - Separating abstracted test cases and their relationships, and increasing modularity
- Relationship with NFR
  - Defining quality characteristics of test suite
- Modeling techniques / Patterns
  - Extracting and defining patterns of abstracted test cases
- Notations / meta-modeling
  - Establishing meta models, symbols, vocab and grammar for describing test architecture

# What to call abstract test cases

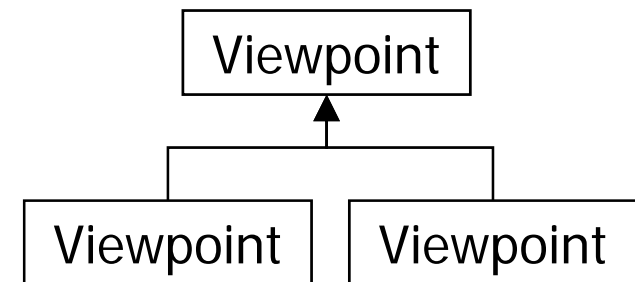
---

- There are several similar concepts of abstract test cases, but these are not perfect concepts
  - Test level and test type
    - » Integrated concept is necessary
  - Test objective (from U2TP etc.)
    - » This is just a relationship and can't be hierarchical (in U2TP definition)
    - » Experts sometimes uses test objectives as goal or purpose of software testing
  - Hierarchical equivalence classes (from classification tree method)
    - » Equivalence classes and relationships can't be described in the same model
  - Test sub-processes (from ISO/IEC DIS 29119)
    - » This sounds a management word
  - Features to be tested (from IEEE std. 829-2008, 29119 etc.)
    - » Meaning of "feature" varies among experts as functions, test items and quality characteristics
  - Test conditions
    - » This usually doesn't mean hierarchical

# The concept of “Viewpoint”

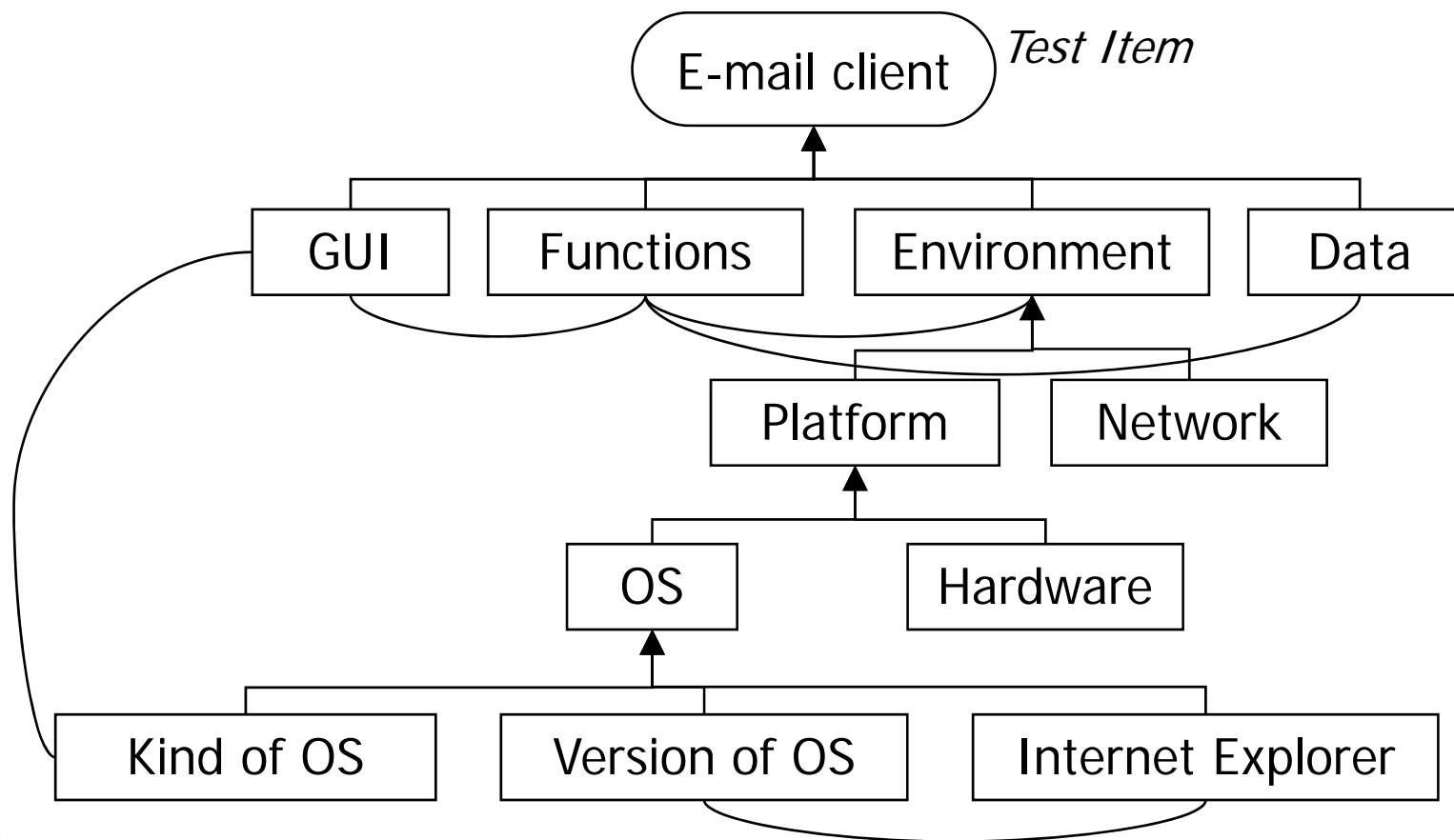
---

- What are requirements of the concept of abstract test cases
  - Shall be abstract test cases
    - » Would be better to be described as a box
  - Shall be an integrated concept of test levels and test types
    - » Test levels means (all or part of ) test items
    - » Test types means behavior as quality characteristics, e.g. performance testing
    - » Test types also means test conditions, e.g. load testing and configuration testing
  - Shall be hierarchical
  - Shall be able to have relationships each other
  - Shall mean test condition at leaf (lowest) level
- This research calls this concept as “Viewpoint”
  - Viewpoint is abstract test cases
  - Viewpoint is an integrated concept of (all or part of) test item, quality characteristics and test condition
  - Viewpoint is hierarchical
  - Viewpoint can have relationships each other
  - Leaf viewpoint means test condition



# Example of part of Viewpoint-based test architecture

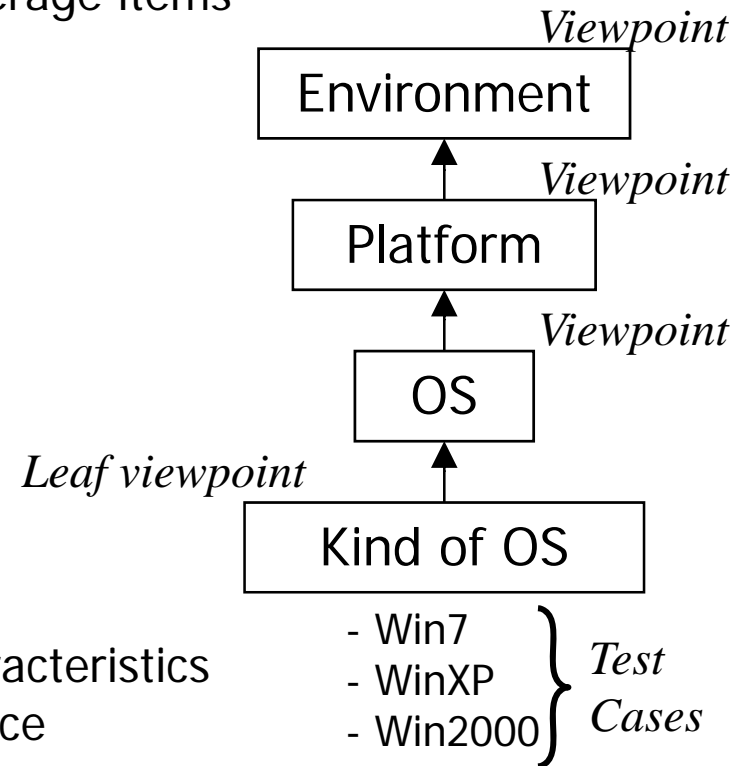
*This research defines NGT: Notation for Generic Testing*



# Viewpoint as abstract test cases

---

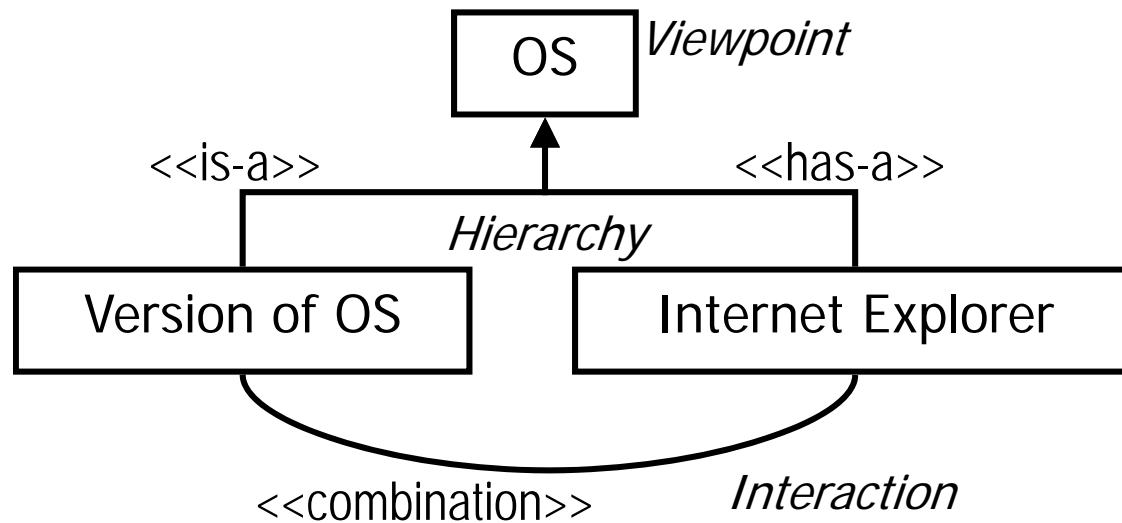
- Test cases has test parameters and values
  - ex) parameter: Kind of OS, values: Win7, WinXP, Win2000
  - Test parameters are also called as test conditions and test values are also called as test coverage items
  - Test cases consists of test values
- Viewpoints are abstract test cases
  - Leaf viewpoints means test parameters
  - Viewpoints don't express any test values or test cases
  - Viewpoints can have hierarchically abstract viewpoint
    - » Like classification trees or class diagrams
  - Viewpoints can abstract test conditions, test items and quality characteristics such as load, configuration and performance



# Relationships of Viewpoints

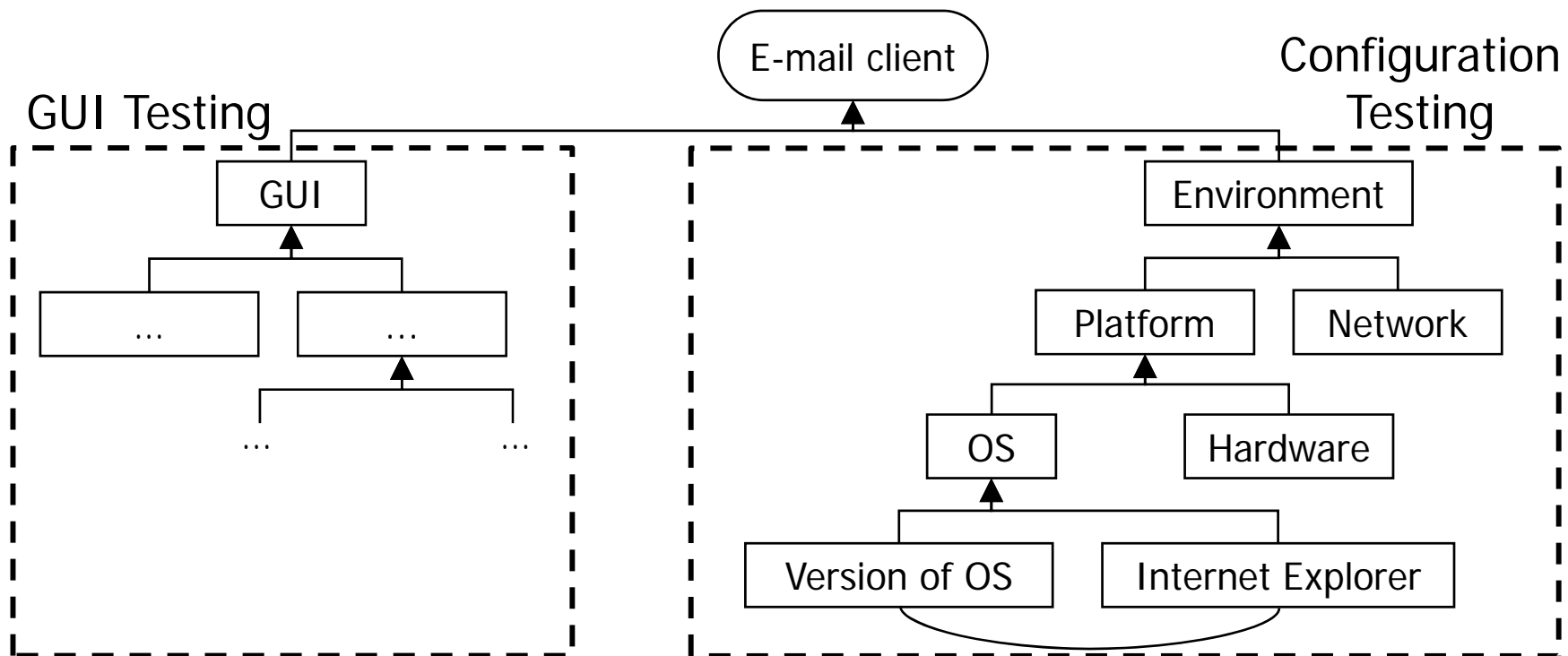
---

- Viewpoints have two fundamental relationships
  - Hierarchy relationships
    - » Detail a viewpoint step by step to reach test coverage item with a straight line
    - » Have several types such as is-a, has-a, cause-effect, object-attribute
  - Interaction relationships
    - » Connect test viewpoints to test combination of viewpoints with a curved line
    - » Have several types such as combination (needs combinatorial testing) etc.
- Types of relationships can be expressed as "<<stereotype>>"



# The Concept of "Test Container"

- For grasping a big picture, several Viewpoints can be organized into one large dotted-line box, "Test Container"
  - Test Container can be test levels and test types in huge test suite

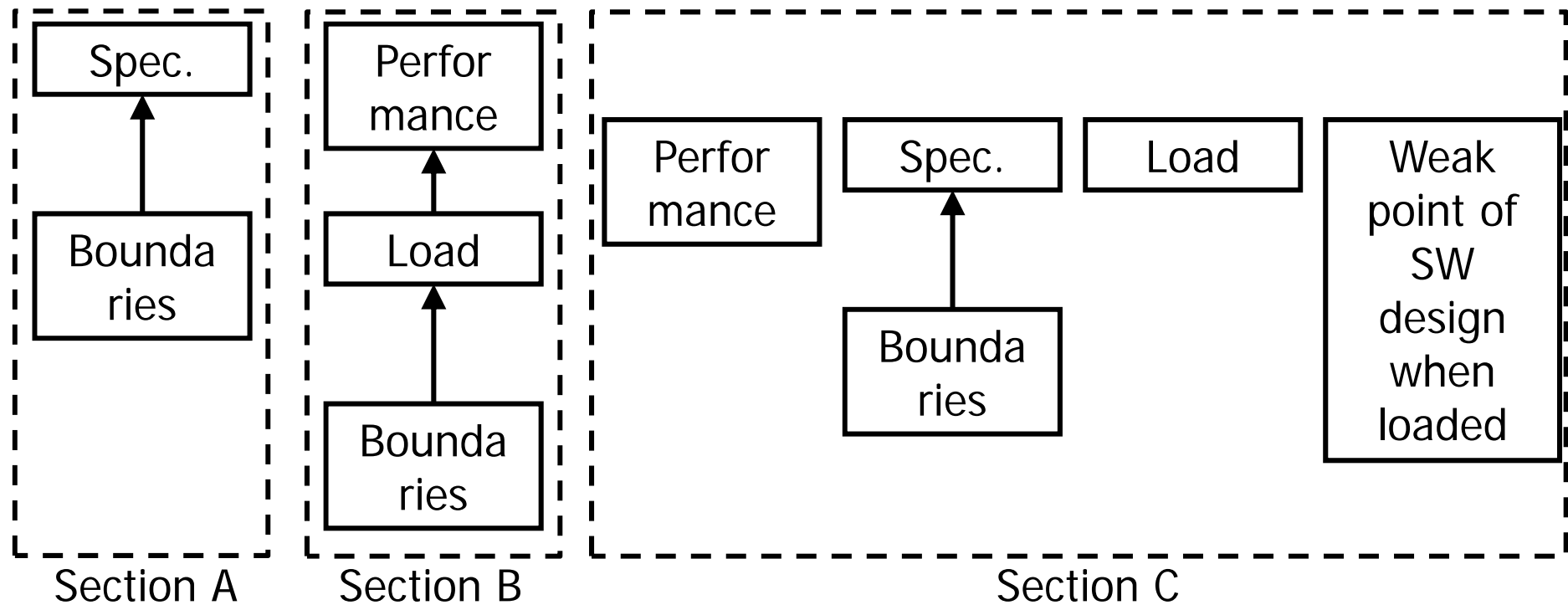




# Meanings of test levels/types depend on organizations

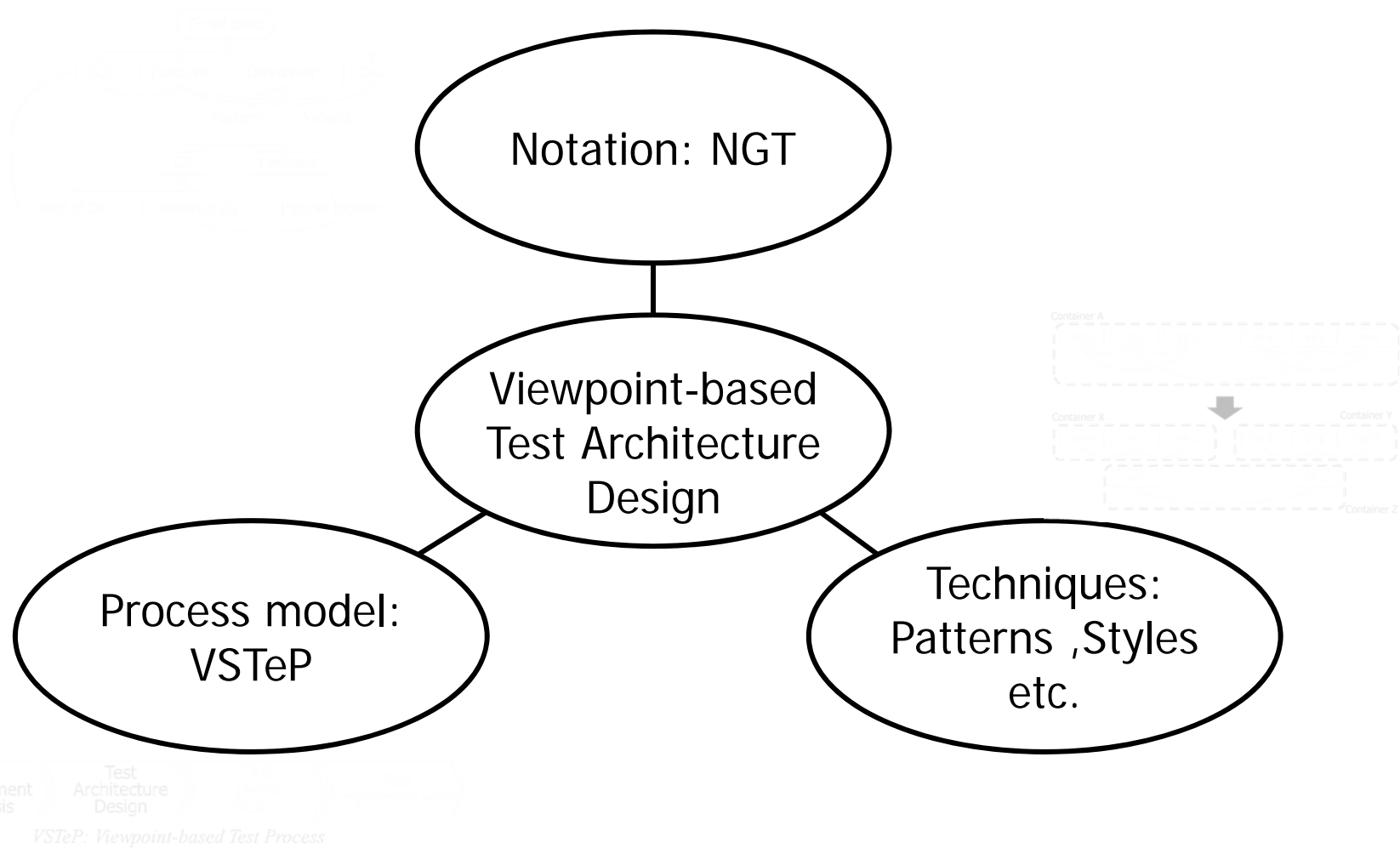
---

- Viewpoints make it clear for organizations what actually to test though they use the same name of test levels/types
  - This is an example of viewpoints for performance testing by neighbor section in the same organization



# Overview of this research

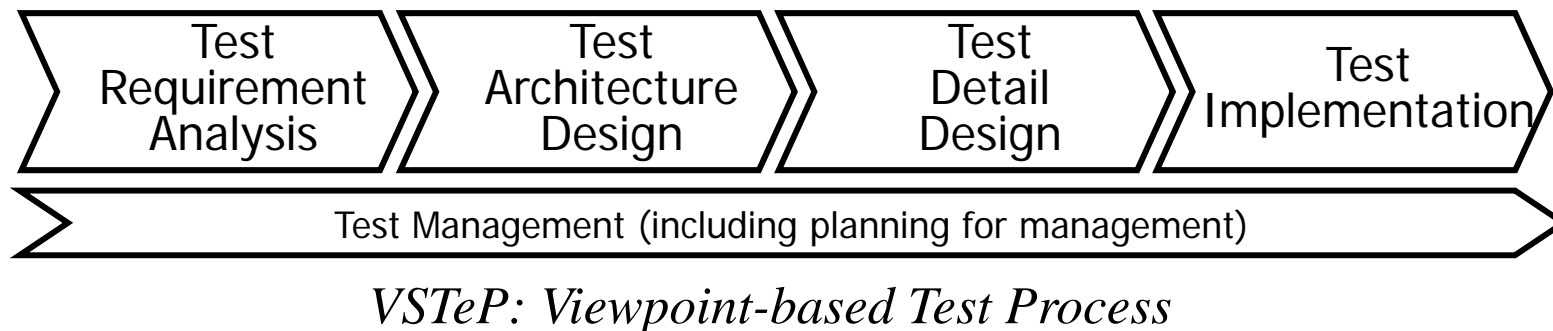
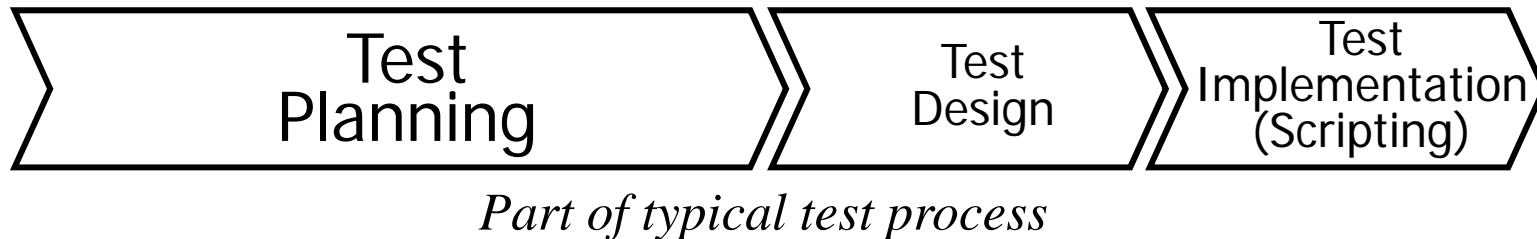
---



# VSTeP: Viewpoint-based Test Process

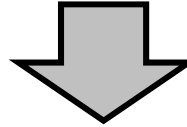
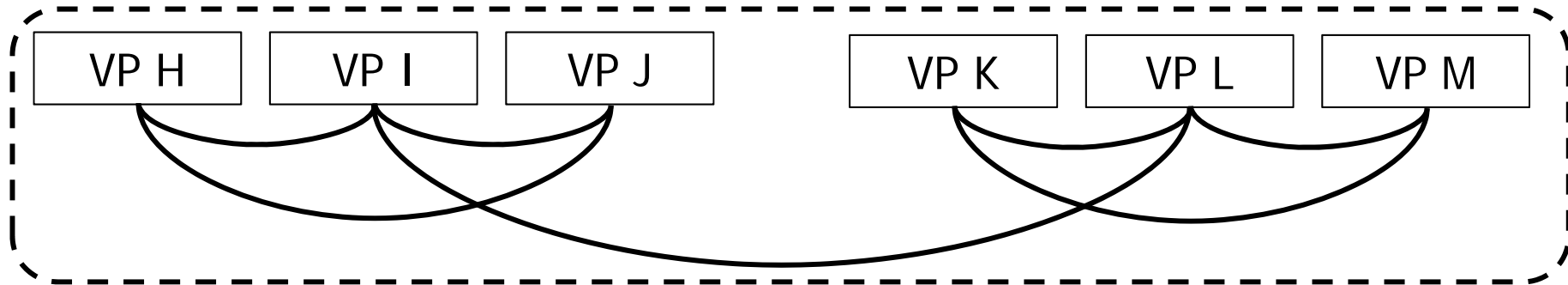
---

- VSTeP is a process model for Viewpoint-based test architecture design

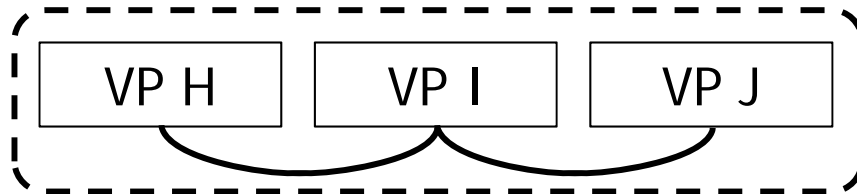


# Example of test architecture patterns: Cluster Dividing

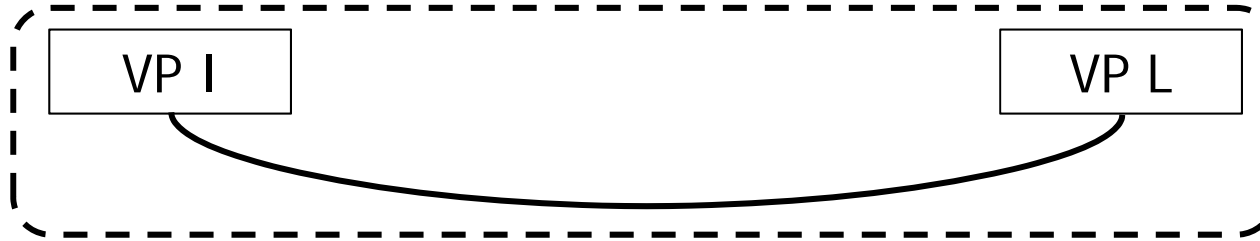
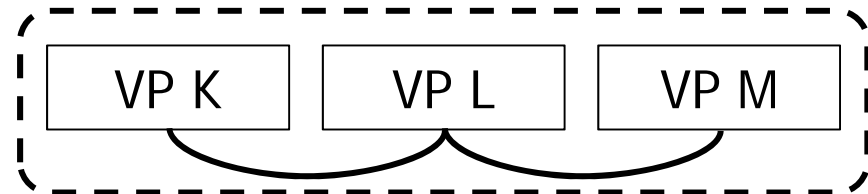
Container A



Container X



Container Y

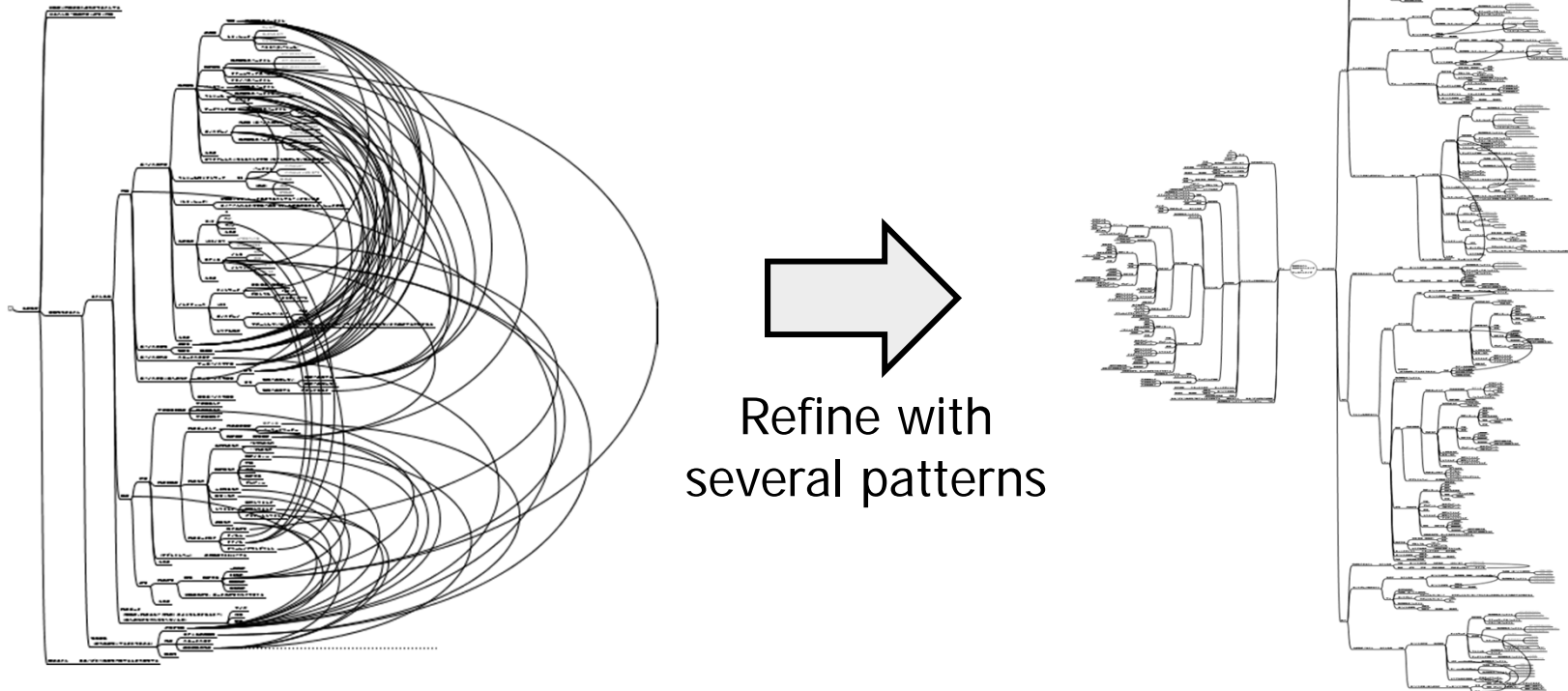


Container Z

# Example of test architecture refinement

---

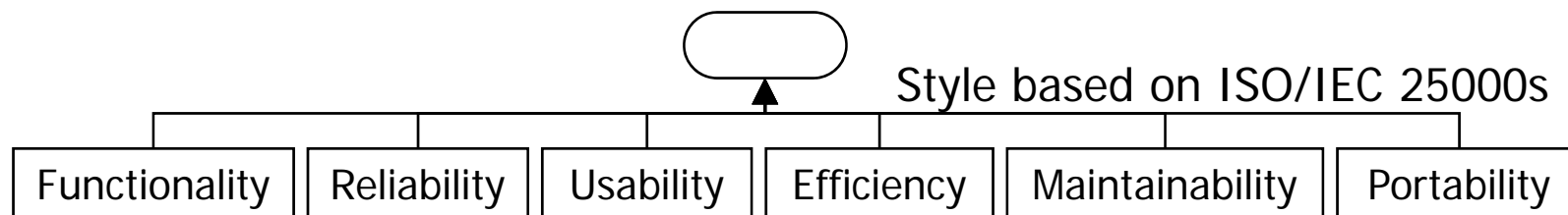
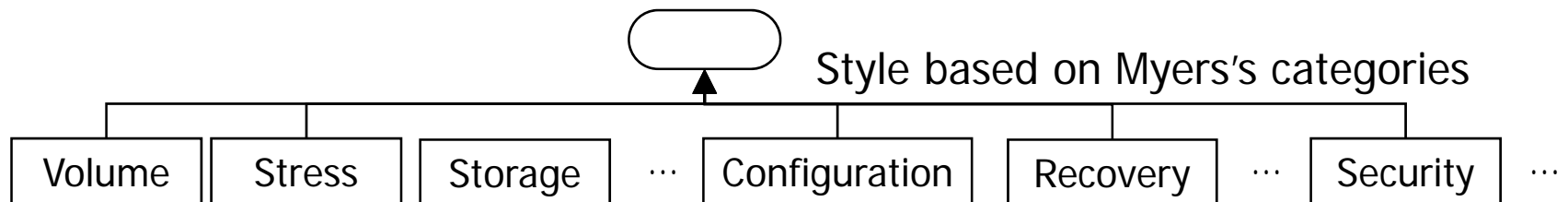
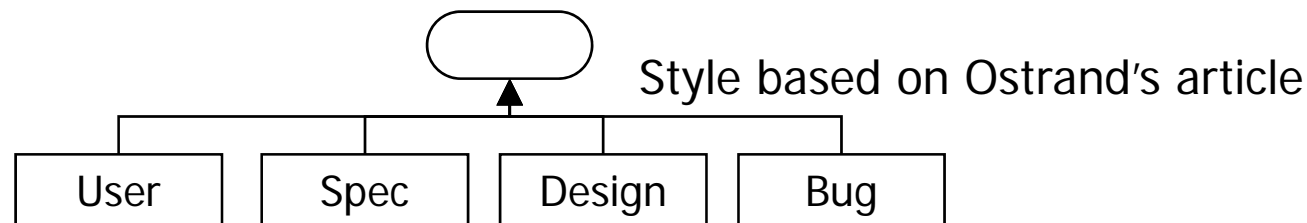
- Several patterns can refine test architecture
  - Both are the same meaning model of semi mission critical software
  - I refined the left model to the right model using several patterns
  - The right model looks easy to be divided into test (sub) levels and types



# Example of Test Architecture Styles in system testing

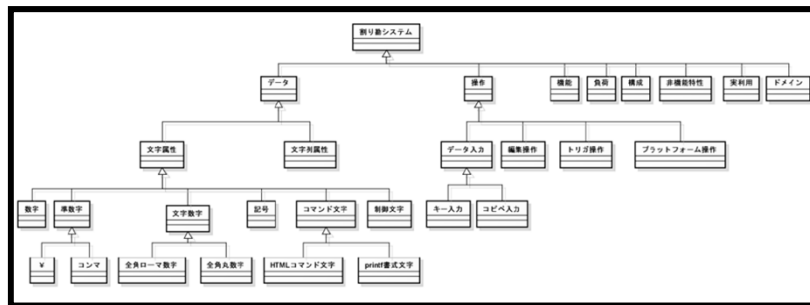
---

- There can be various test architecture styles



# Quality attributes of test suite

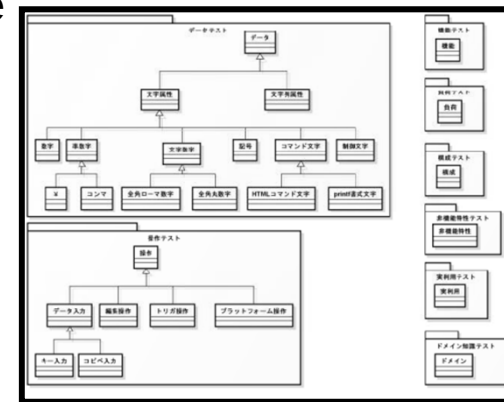
- Test architecture depends on required quality attributes of test suite
  - Test suite can have its own quality attribute if test suite is artifact
  - Ex) Maintainability of test suite
  - It doesn't mean testing of quality attribute such as ISO/IEC 25000s/9126s



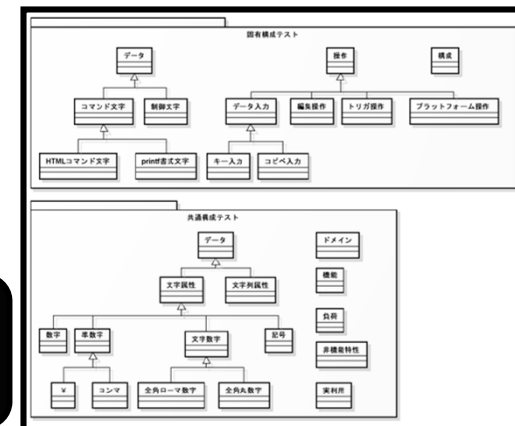
Test architecture of small calculator



Maintainability considered



Simply divided



# Relationships to international standard

---

- For ISO/IEC DIS 29119 part 2 and IEEE std. 829-2008
  - This research could be add-on for them as design of test strategy and definition of test levels
    - » 6.2.3.5 “Design Test Strategy (TP5)” c) of 29119-2: A Test Strategy (comprising choices including test phases, test types, features to be tested, test techniques and test completion criteria) shall be designed that considers employing the treatment actions for those risks addressed in the previous task and takes into account the requirements of the Organizational Test Strategy.
    - » 8.2.1 “(MTP Section 2.1) Test processes including definition of test levels” of 829-2008: Identify the number and sequence of levels of test. There may be a different number of levels than the example used in this standard (component, component integration, system, and acceptance).
- For ISO/IEC 25000s
  - This research could make a possibility of new types of quality attributes
- For ISO/IEC 12207
  - This research might be harmonized to if test suite is recognized as artifact

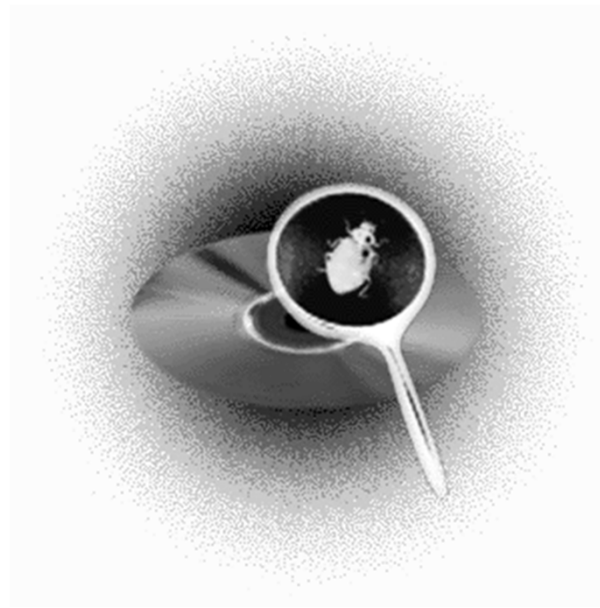


# Conclusion

---

- Research area on test architecture is necessary
- I proposed Viewpoint-based Test Architecture Design
  - Viewpoints / Relationships / Containers / Notation: NGT / Process model: VSTeP / Patterns / Refinement / Styles / Quality attributes of test suite
- Viewpoint-based Test Architecture Design could contribute several standards
  - ISO/IEC DIS 29119s / IEEE std. 829-2008 / ISO/IEC 25000s / ISO/IEC 12207 ?
- Viewpoint-based Test Architecture Design can make several research area such as:
  - Test architecture patterns
  - Test architecture styles
  - Quality characteristics of test suite
  - Detail test architecture design process
  - Better notations
  - Architecture-level reuse / product lines of test suite
  - Metrics for test architecture ?

*Let's start research on test architecture!*



NISHI, Yasuharu  
<http://blues.se.uec.ac.jp/>  
Yasuharu.Nishi@uec.ac.jp

© NISHI, Yasuharu