Software Engineering for Distributed Systems Group
Georg-August-Universität Göttingen

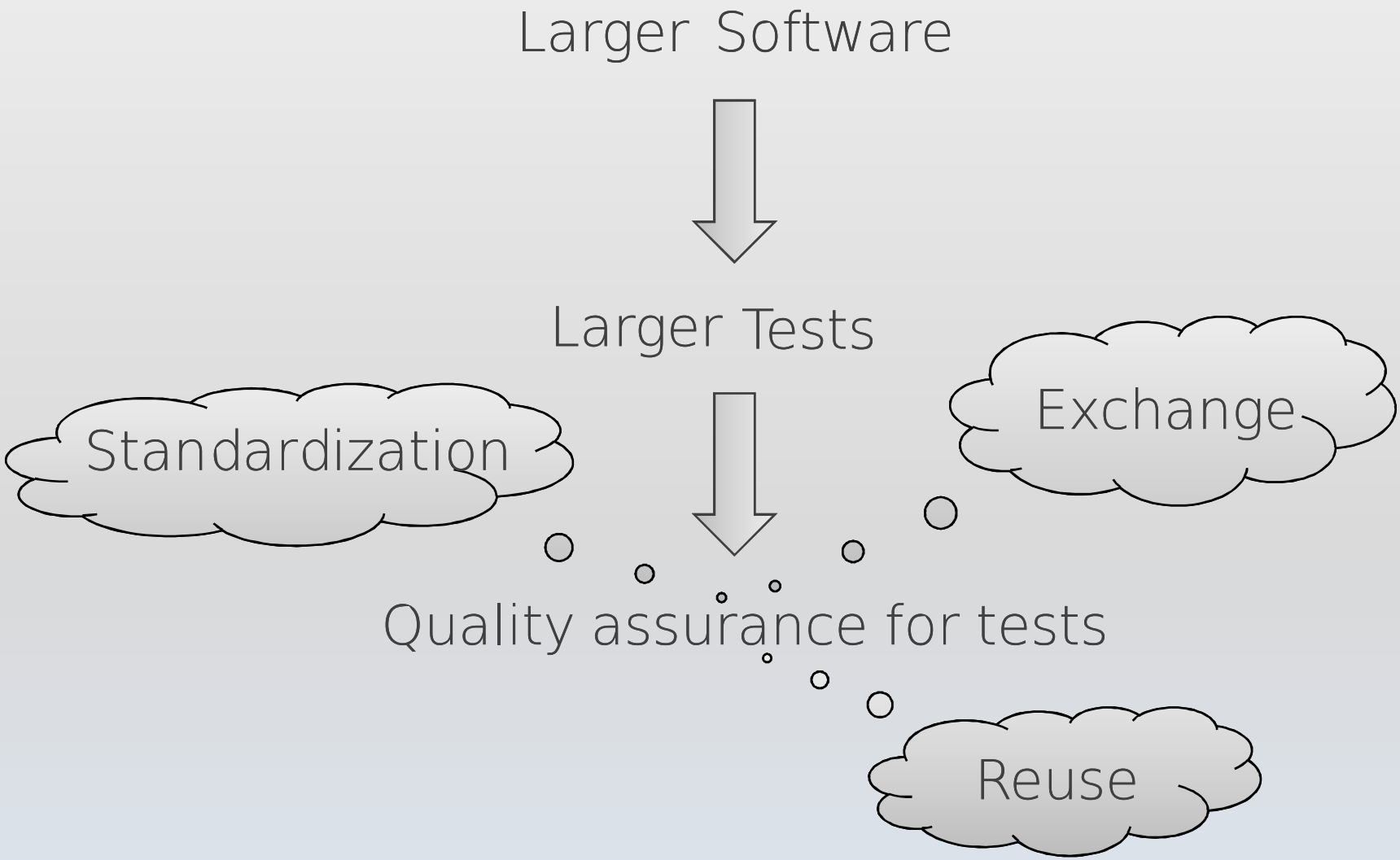
Systematic Quality Assurance for Test Specifications

Philip Makedonski
Jens Grabowski

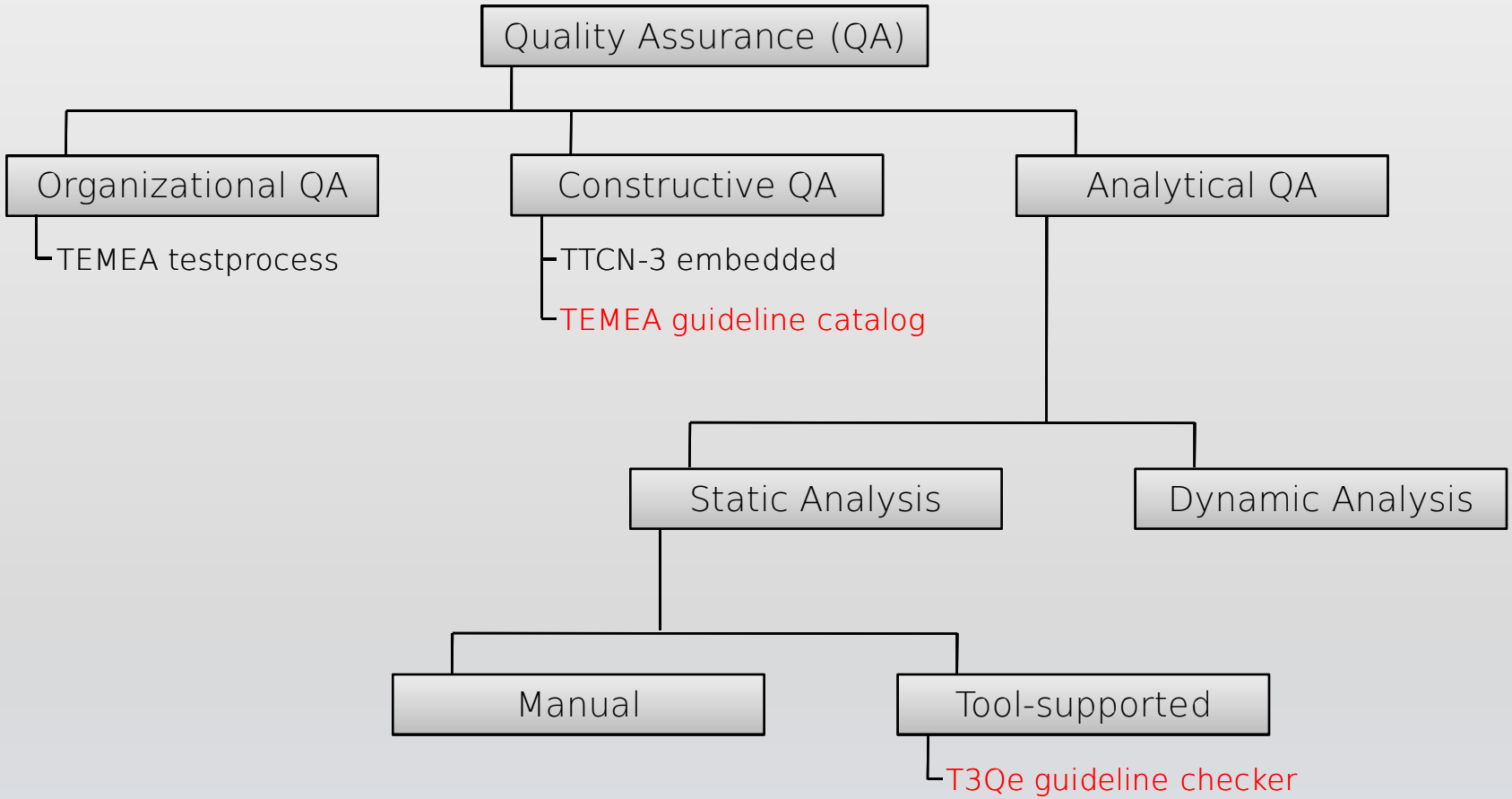
Overview

- Motivation
- Systematic Quality Assurance for Tests
  - Guidelines
  - Automated Checking
- Integrated Quality Assurance for TTCN-3
- Summary and Outlook

Motivation


Larger Software
Larger Tests
Standardization
Exchange
Quality assurance for tests
Reuse

Systematic Quality Assurance for Tests



What is wrong with this picture?

Format!

```
module workshopExample language

    type component c1 {
        port streamPort a;
        port messagePort b;
        timer c;
        var integer d;
    }
```

```
10      testcase t1(
11          float x,
12          someObscureType y
13      )
14      runs on c1 { /*some complex behavior here*/
15          cont {
16              par {
17                  seq {
18                      cont { /**/
19                          a.value := x;
20                      }
21                      cont { /**/
22                          b.value := x + 2;
23                      }
24                  }
25                  cont { /**/
26                  }
27              }
28          }
29      }
```

What is wrong with this picture?

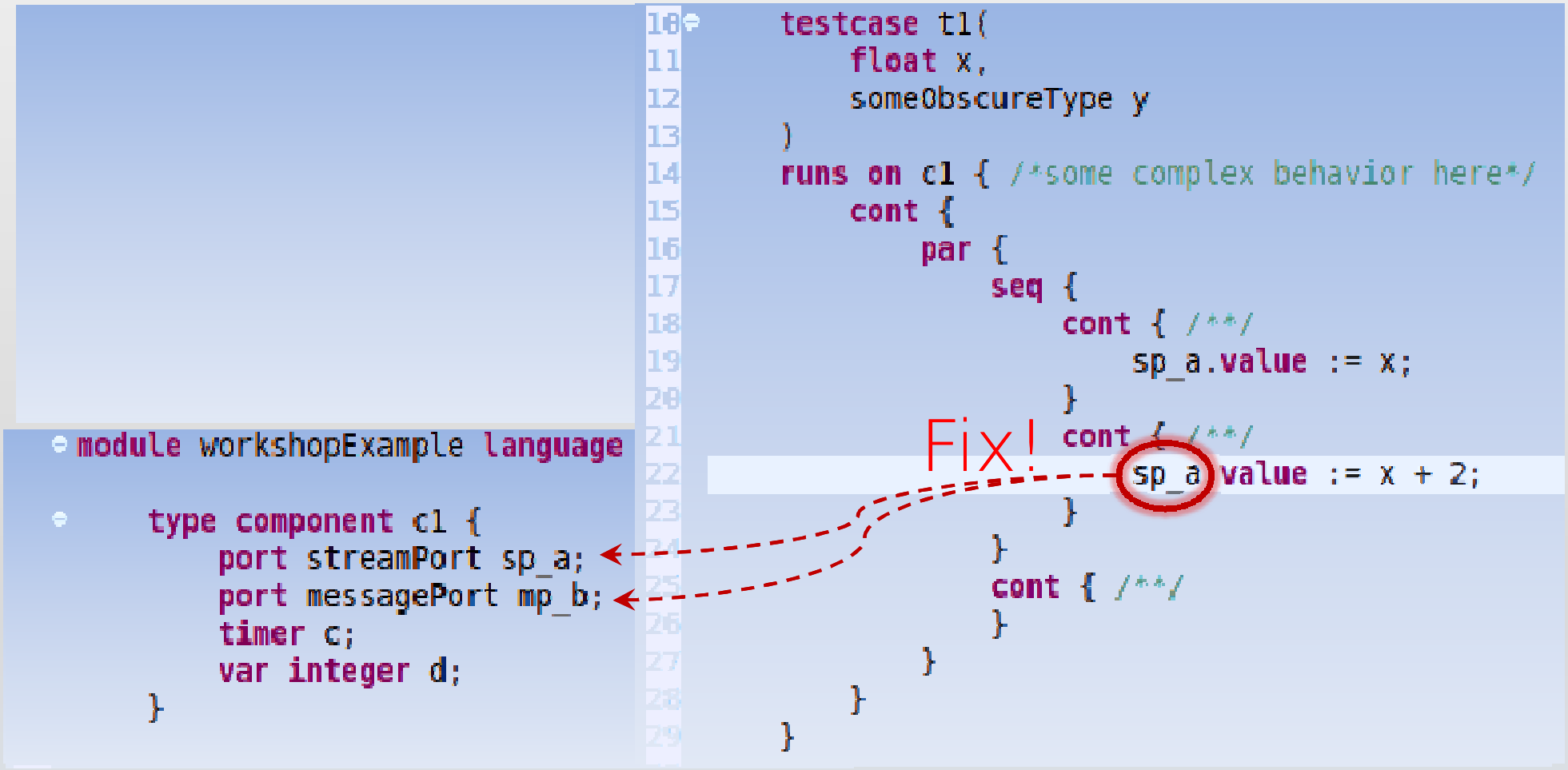

```
module workshopExample language

    type component c1 {
        port streamPort sp_a;
        port messagePort mp_b;
        timer c;
        var integer d;
    }
```

```
    testcase t1(
        float x,
        someObscureType y
    )
    runs on c1 { /*some complex behavior here*/
        cont {
            par {
                seq {
                    cont { /**/
                        sp_a.value := x;
                    }
                    cont { /**/
                        mp_b.value := x + 2;
                    }
                }
                cont { /**/
                }
            }
        }
    }
```

Rename!

What is wrong with this picture?



Guidelines: Definition

- Rules and recommendations for source code
  - Respect human aspects
  - Often in the form of
    - metrics with thresholds
    - structural and semantic properties
- Based on
  - Practical experience
  - Project and domain specific requirements

Guidelines: Categories

- Naming conventions
- Code formatting
- Code documentation
- Code style
- etc.

Guidelines: TTCN-3 / TTCN-3 embedded

G1: Conformance to Naming Conventions

Description:

- Rules for systematic naming of identifiers.
- Consist of contexts and rules.
- Improve readability, understandability, traceability, and changeability.
- May help avoiding errors.

Properties:

- project specific,
- configurable,
- applicable for TTCN-3, TTCN-3 embedded

Guidelines: TTCN-3 / TTCN-3 embedded

G1: Conformance to Naming Conventions (continued)

Example instantiations:

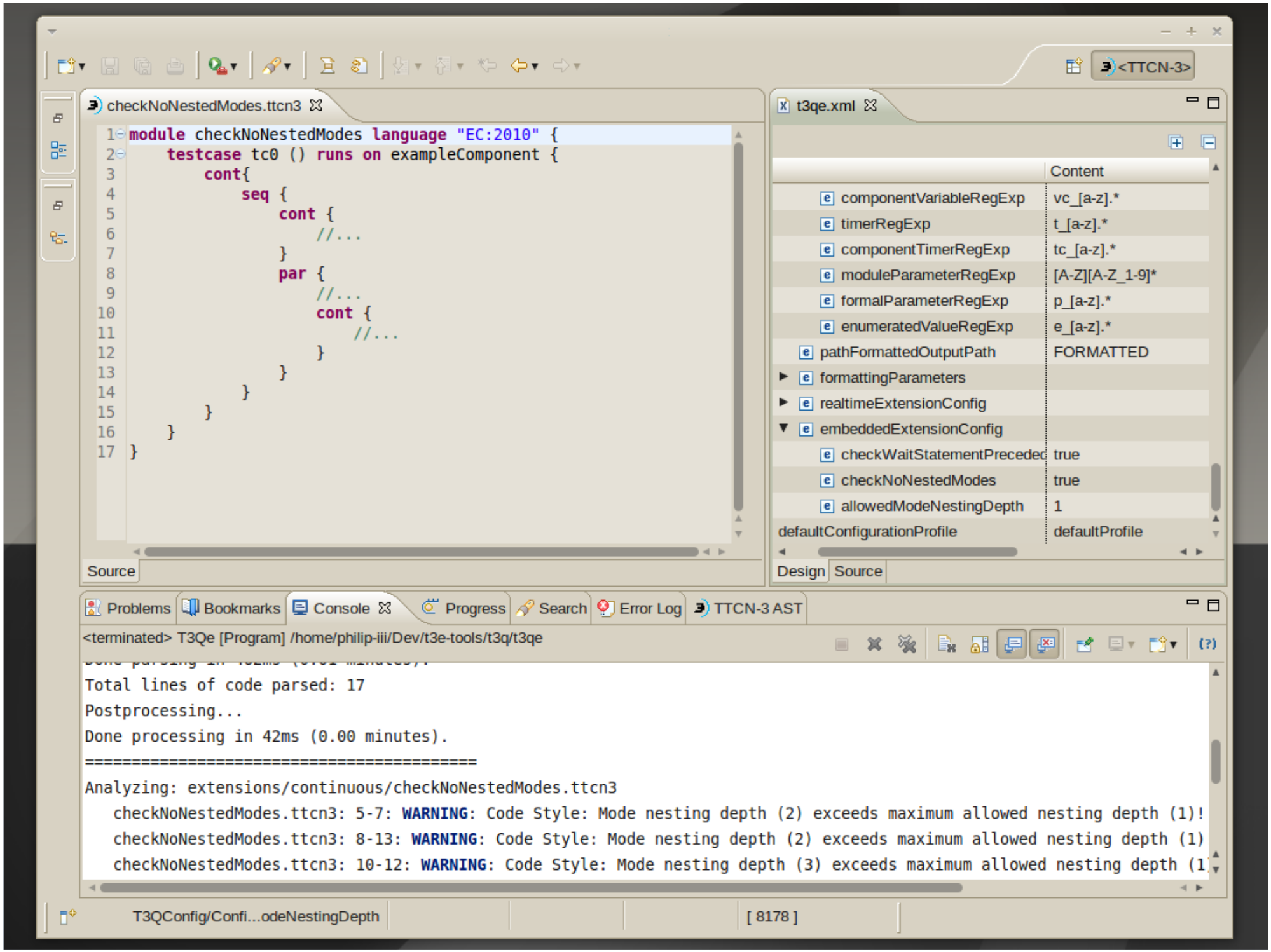
- local constant: cl_[a..z,0..9]*
- constant: c_[a..z,0..9]*
- function: f_[a..z,0..9]*
- external function: fx_[a..z,0..9]*
- message port: mp_[a..z,0..9]*
- stream port: sp_[a..z,0..9]*
- sequential mode: seq_[a..z,0..9]*
- parallel mode: par_[a..z,0..9]*

Guidelines: Catalog

- 23 guidelines
  - Reference ID
  - Description including implications
  - Properties
  - Example instantiations

Guidelines: Checking – T3Qe

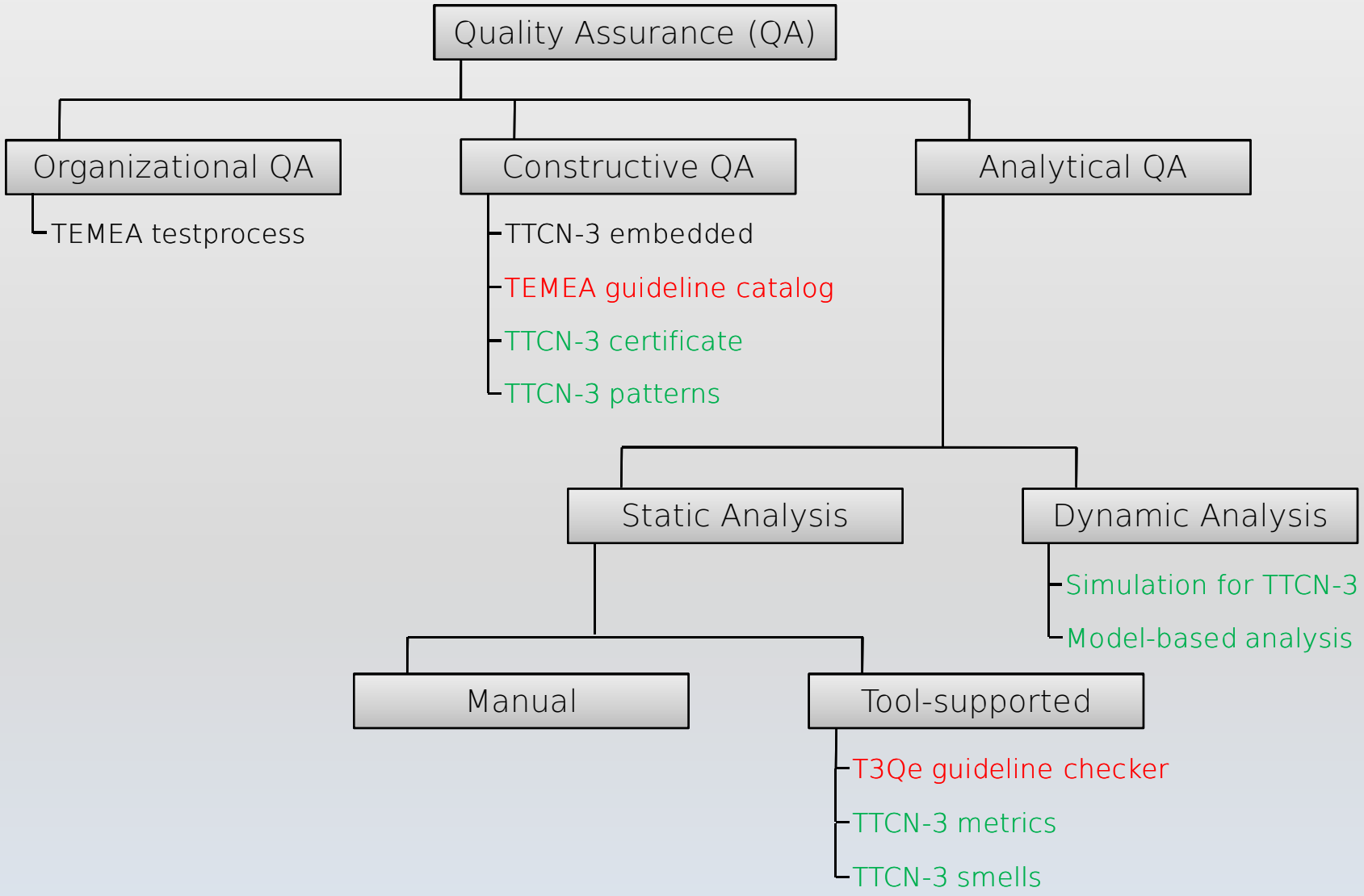
- Static analysis guideline checking
  - Implements guidelines from the catalog
  - Supports TTCN-3 v4.1.1 and TTCN-3 embedded (draft)
- Project specific profiles
- Code formatting
- Cross-platform
- Open source (Eclipse Public License)


checkNoNestedModes.ttcn3
module checkNoNestedModes language "EC:2010" {
testcase tc0 () runs on exampleComponent {
cont{
seq {
cont {
//...
}
par {
//...
cont {
//...
}
}
}
}
}
}
Source
t3qe.xml
Content
componentVariableRegExp
vc_[a-z].*
timerRegExp
t_[a-z].*
componentTimerRegExp
tc_[a-z].*
moduleParameterRegExp
[A-Z][A-Z_1-9]*
formalParameterRegExp
p_[a-z].*
enumeratedValueRegExp
e_[a-z].*
pathFormattedOutputPath
FORMATTED
formattingParameters
realtimeExtensionConfig
embeddedExtensionConfig
checkWaitStatementPrecedec
true
checkNoNestedModes
true
allowedModeNestingDepth
1
defaultConfigurationProfile
defaultProfile
Design
Source
Problems
Bookmarks
Console
Progress
Search
Error Log
TTCN-3 AST
<terminated> T3Qe [Program] /home/philip-iii/Dev/t3e-tools/t3q/t3qe
Total lines of code parsed: 17
Postprocessing...
Done processing in 42ms (0.00 minutes).
Analyzing: extensions/continuous/checkNoNestedModes.ttcn3
checkNoNestedModes.ttcn3: 5-7: WARNING: Code Style: Mode nesting depth (2) exceeds maximum allowed nesting depth (1)!
checkNoNestedModes.ttcn3: 8-13: WARNING: Code Style: Mode nesting depth (2) exceeds maximum allowed nesting depth (1)
checkNoNestedModes.ttcn3: 10-12: WARNING: Code Style: Mode nesting depth (3) exceeds maximum allowed nesting depth (1)
T3QConfig/Confi...odeNestingDepth
[8178]

Overview

- Motivation
- Systematic Quality Assurance for Tests
  - Guidelines
  - Automated Checking
- Integrated Quality Assurance for TTCN-3
- Summary and Outlook

Systematic Quality Assurance for Tests in TTCN-3



Further Tooling for TTCN-3

- Improvement by refactorings
  - TRex
    - Editor
    - IDE
    - TTCN-3 Front-end

- Documentation
  - T3D
    - Uses TTCN-3 Documentation Comment Specification
    - Generates navigable HTML documentation

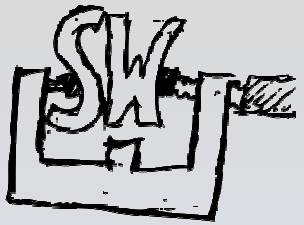
Summary and Outlook

- Summary
  - Systematic quality assurance for tests
  - Guidelines
  - Automated checking
  - Integrated quality assurance for TTCN-3
- Outlook
  - Dynamic analysis for TTCN-3 embedded
  - Additional guidelines

Contact

Philip Makedonski

Software Engineering for Distributed Systems Group
Institute of Computer Science
Georg-August-Universistät Göttingen



makedonski@informatik.uni-goettingen.de
www.swe.informatik.uni-goettingen.de

References (1/4)

- TTCN-3 guidelines
  - Din, G. Vega, D., Schieferdecker, I.
    - Automated Maintainablity of TTCN-3 Test Suites Based *O*n Guideline Checking, IFIP 2008

- TTCN-3 certification
  - German Testing Board
    - http://german-testing-board.info/de/ttcn3_certificate.shtm

References (2/4)

- TTCN-3 patterns
  - Vouffo-Feudijo, A., Schieferdecker, I.
    - Test Patterns with TTCN-3, TestCom/FATES 2004
  - Neukirchen, H.
    - Languages, Tools and Patterns for the Specification of Distributed Real-Time Tests, Dissertation, 2004

- TTCN-3 smells
  - Neukirchen, H., Bisanz, M.
    - Utilising Code Smells to Detect Quality Problems in TTCN-3 Test Suites, TestCom/FATES 2007
  - Bisanz, M.
    - Pattern-based Smell Detection in TTCN-3 Test Suites, Master's Thesis 2006

References (3/4)

- TTCN-3 metrics
  - Neukirchen, H., Zeiß, B., Grabowski, J.
    - An Approach to Quality Engineering of TTCN-3 Test Specifications, STTT 2008
  - Vega, D., Din, G., Schieferdecker, I.
    - TTCN-3 Test Data Analyzer Using Constraing Programming, ICSENG 2008
  - Vega, D., Schieferdecker, I., Din, G
    - Test Data variance as a Test Quality Measure: Exemplified for TTCN-3, TestCom/FATES 2007
  - Zeiß, B.
    - A Refactoring Tool for TTCN-3, Master's Thesis 2006

References (4/4)

- TTCN-3 dynamic analysis
  - Zeiß, B.
    - Quality Assurance of Test Specifications for Reactive Systems, Dissertation 2010
  - Zeiß, B, Grabowski, J
    - Analyzing Response Inconsistencies in Test Suites, TestCom/FATES 2009
  - Makedonski, P, Neukirchen, H, Grabowski, J
    - Validating the Behavioral Equivalence of TTCN-3 Test Cases, VALID 2009
  - Makedonski, P.
    - Equivalence Checking of TTCN-3 Test Case Behavior, Master's Thesis 2009

Backup Slides

Guidelines: TTCN-3 / TTCN-3 embedded

G2: Limited Nesting Depths

Description:

- Restricts the nesting of certain constructs.
- Consists of context and depth.
- Improve readability, understandability, and reusability.
- May help prevent undesirable behavior.

Properties:

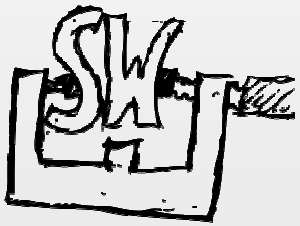
- project specific,
- configurable,
- applicable to TTCN-3, TTCN-3 embedded

Guidelines: TTCN-3 / TTCN-3 embedded

G2: Limited Nesting Depths (continued)

Example instantiations:

- nesting depth limit for alt constructs: 2
- nesting depth limit for altstep constructs: 2
- nesting depth limit for conditionals: 3
- nesting depth limit for mode constructs: 1



Software Engineering for Distributed Systems Group
Georg-August-Universität Göttingen

Systematic Quality Assurance for Test Specifications

Philip Makedonski
Jens Grabowski