t3tools TTCN-3 ETSI World Class Standards 3GPP A GLOBAL INITIATIVE

Dependency and Guideline Analysis for TTCN-3

Steffen Herbold, Philip Makedonski,
Jens Grabowski, Kathrin Becker,
Stefan Kirchner, Benjamin Zeiss

Georg-August-Universität Göttingen, Germany

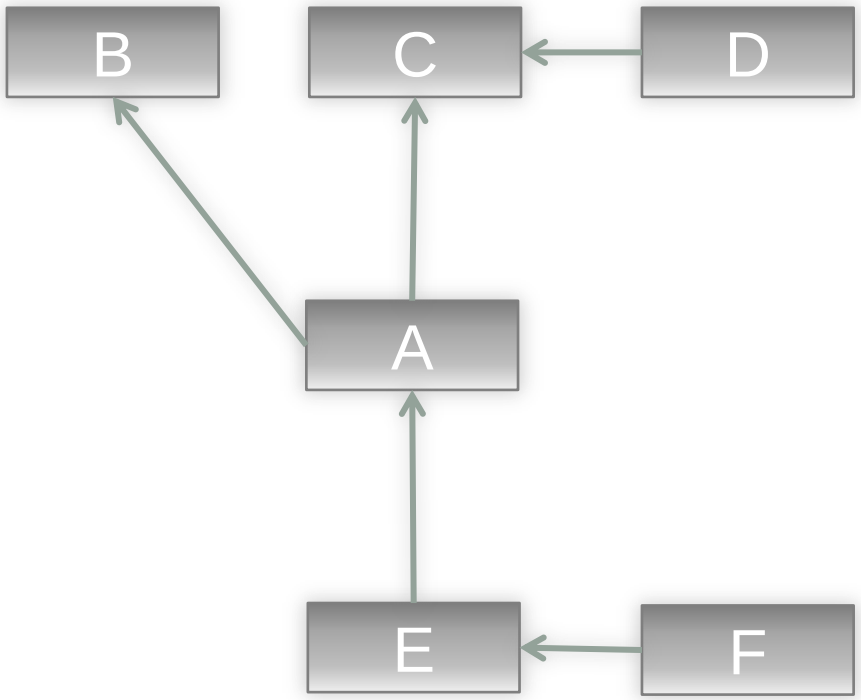
Outline

- Motivation
- Dependency Analysis
- Guideline Analysis
- Tools
- Summary & Outlook

Motivation

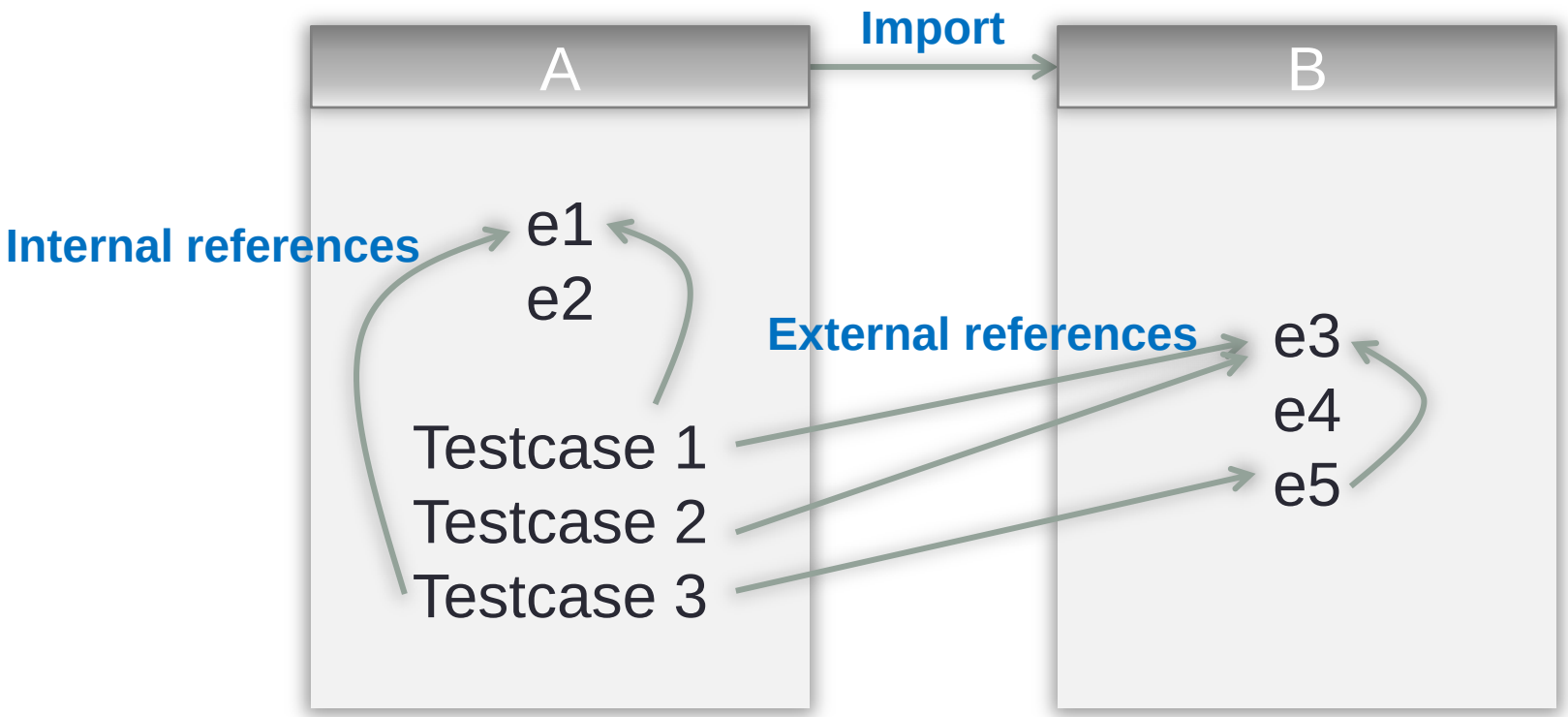
- Increasing test suite size and complexity
- More than 200.000 LOC for next-generation test suites
  - Maintainability?
- Enforcing **guidelines**
  - Prevents mistakes
  - Reduces effort for maintenance and deliveries
- Early dev. version of the ETSI 3GPP LTE/SAE test suite:
  - Approx. 20.000 LOC
  - 411 imports
  - 3361 references
  - 2457 external references
  - 904 internal references
- **Dependencies** promote quality attributes

Motivation: Module Dependencies



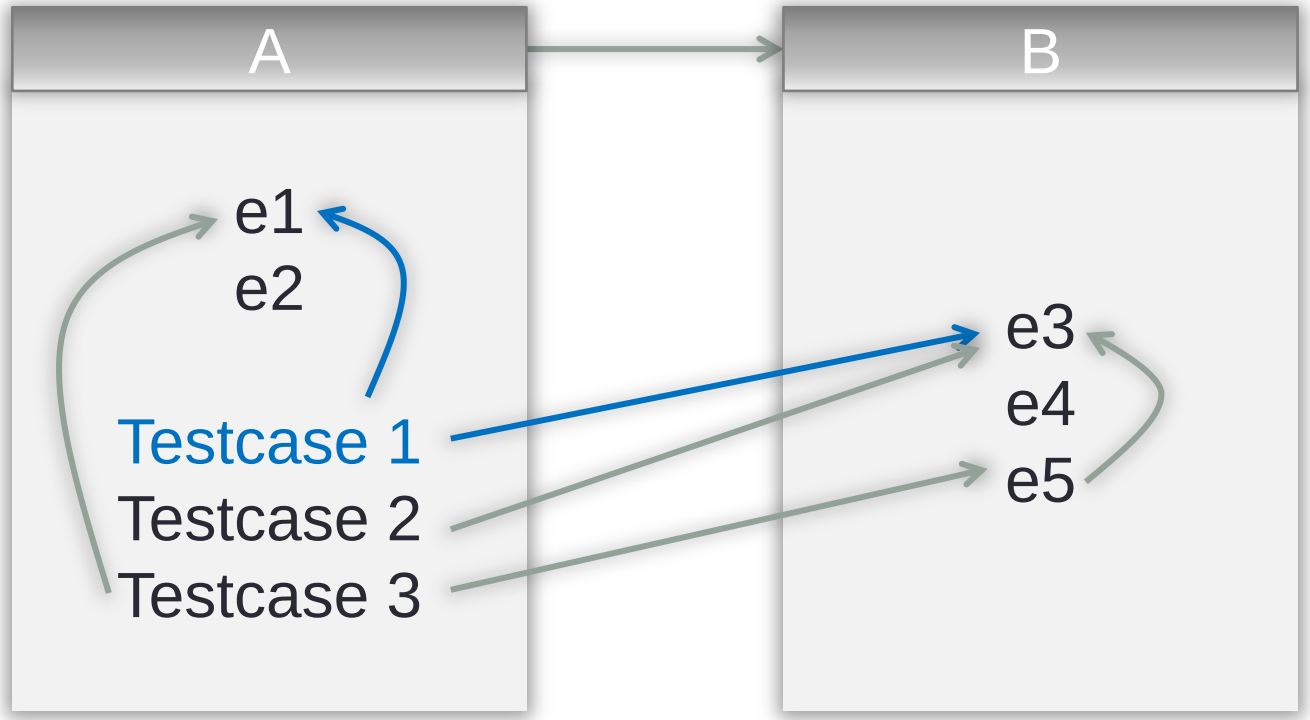
- How does a local change affect the rest of the test suite?
- Are there any superfluous imports?
- What elements are affected by an element freeze?
- Is a module a library?
- Is a module element public, private, or deprecated?

Motivation: Module Dependencies



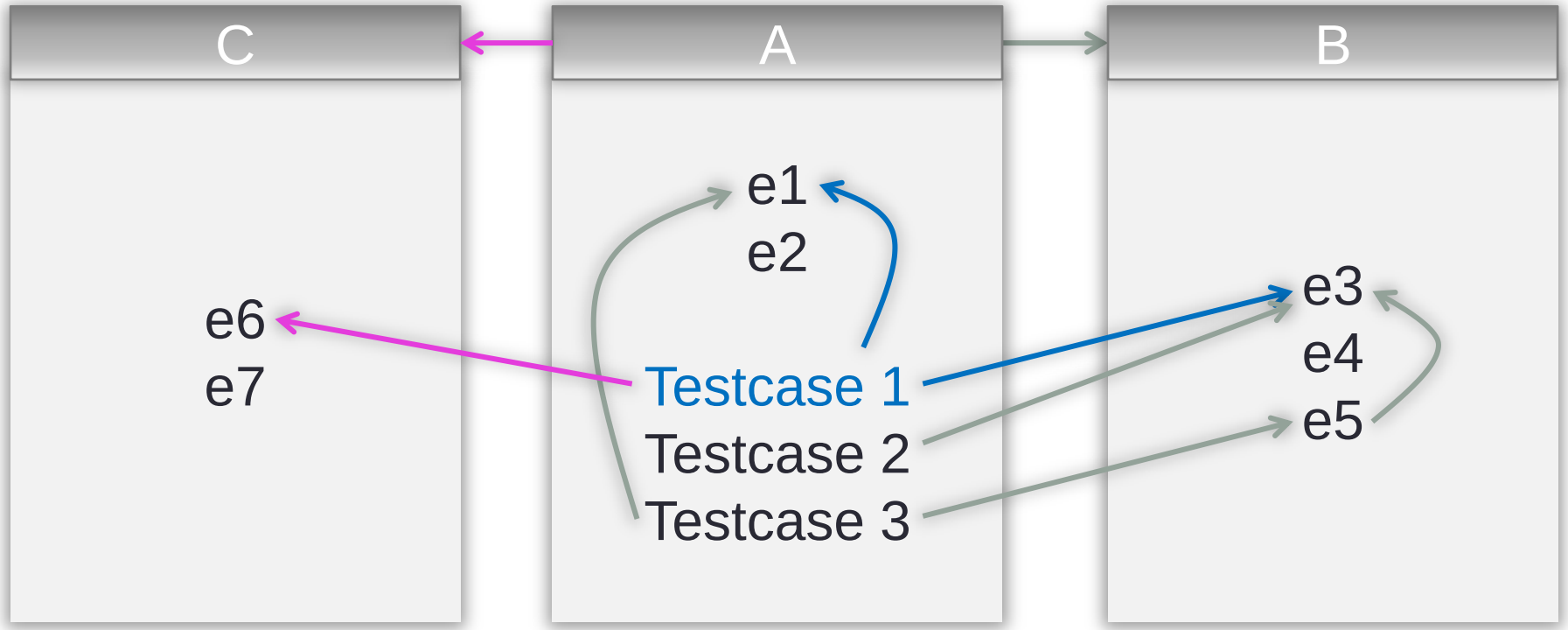
Dependency Analysis

- Local change:
  - Change of test case behavior.



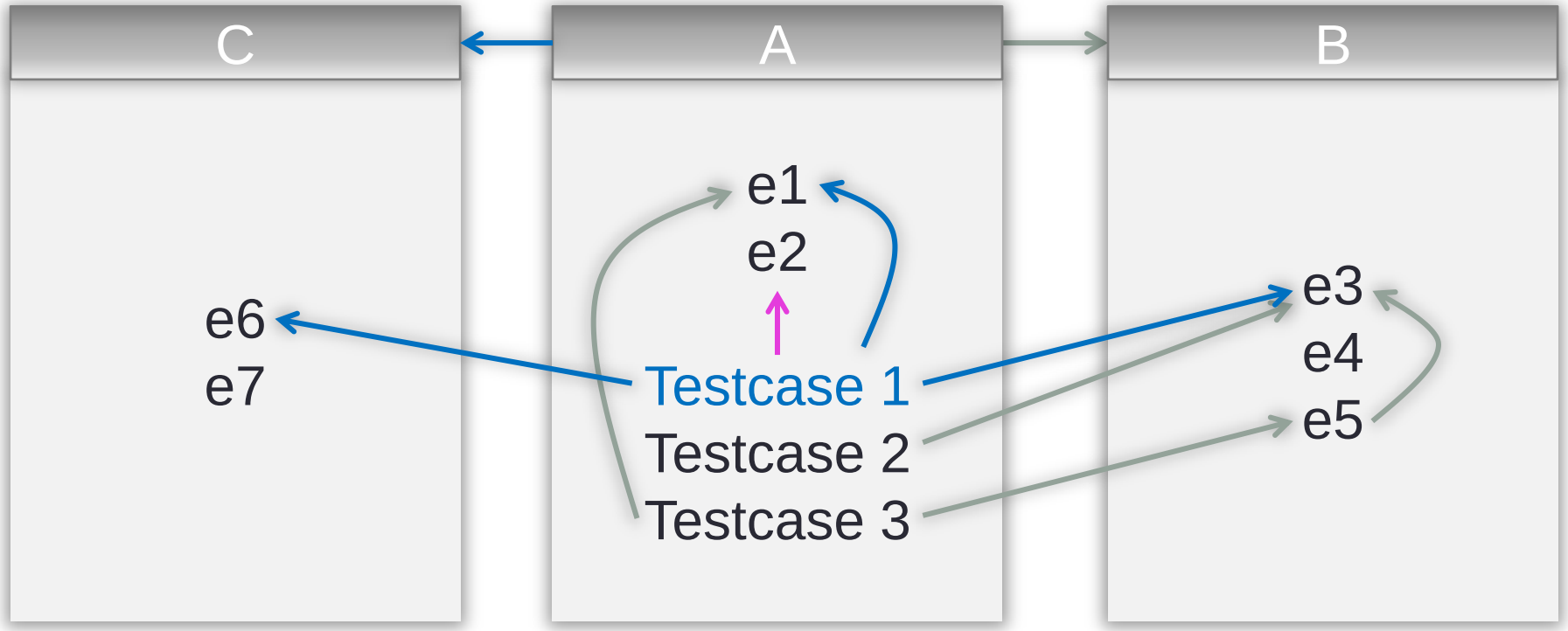
Dependency Analysis

- Local change:
  - Addition of new external dependencies, higher coupling, less reusability



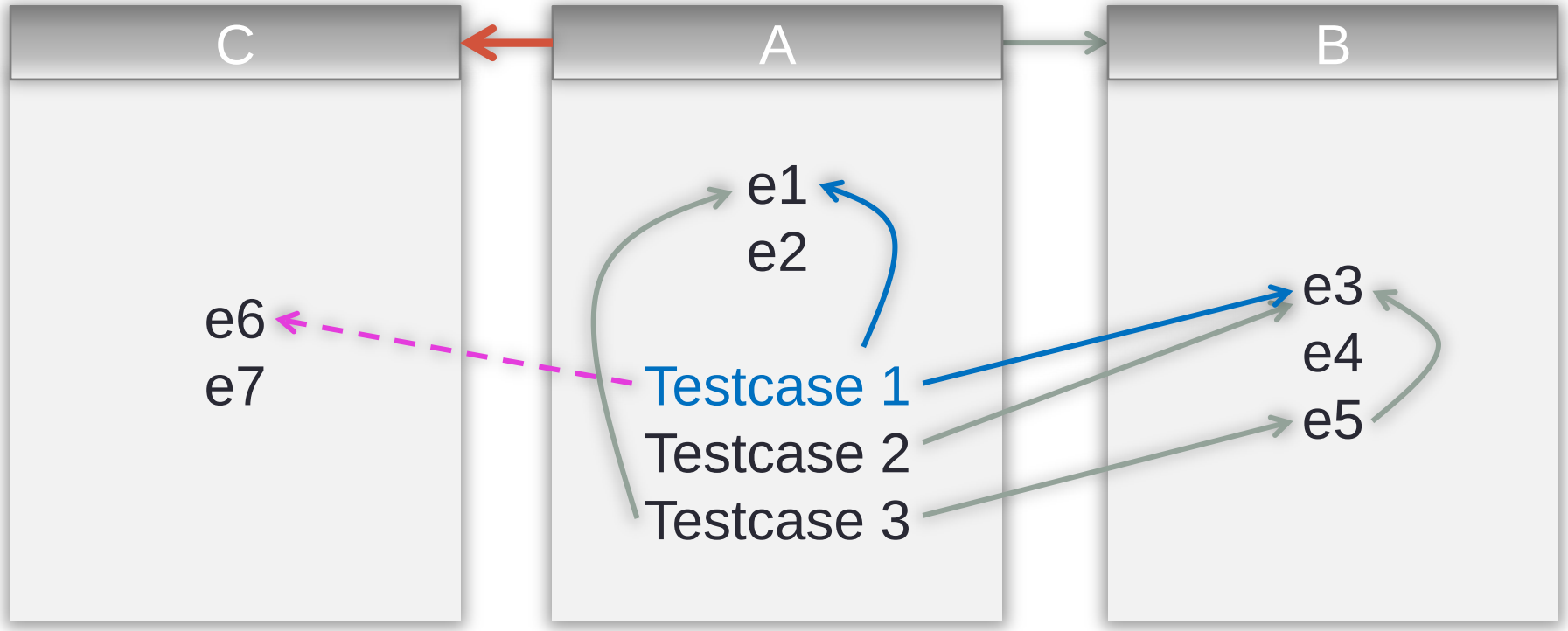
Dependency Analysis

- Local change:
  - Addition of new internal dependencies, higher cohesion



Dependency Analysis

- Local change:
  - Removal of dependencies, less coupling, superfluous imports

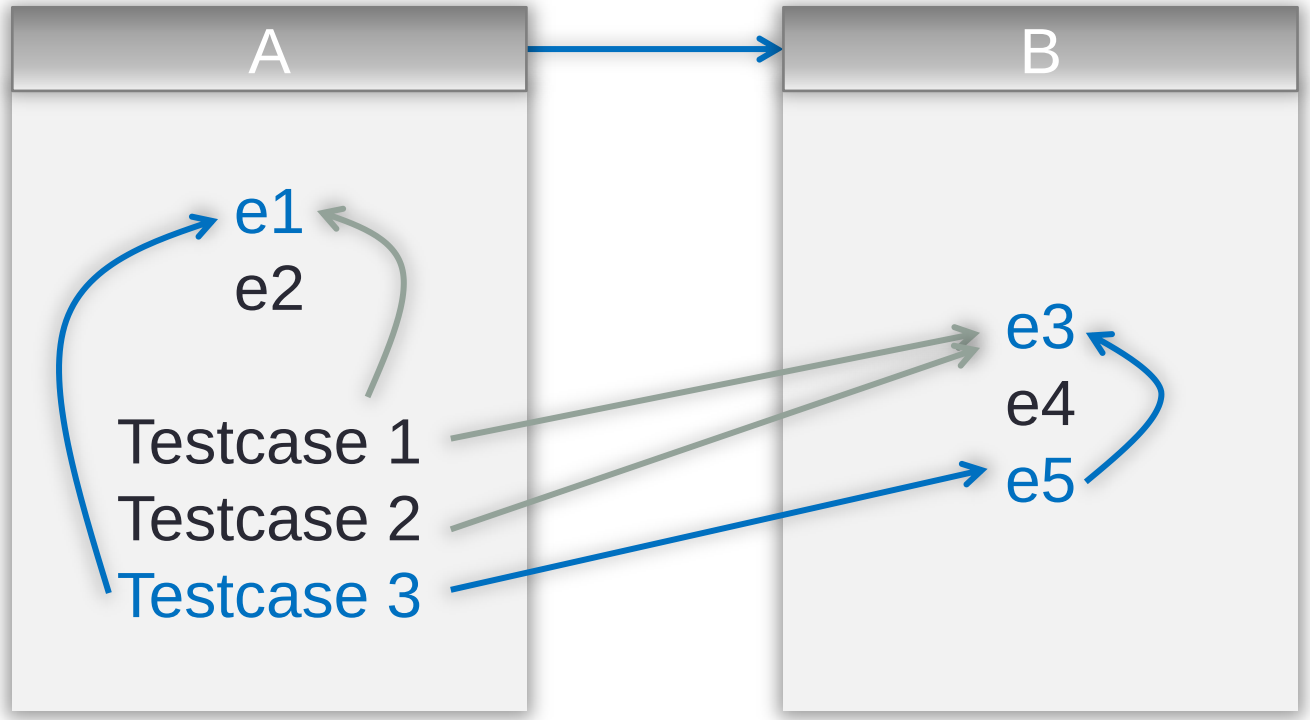


Dependency Analysis

- Local change:
  - To get the big picture, we have to follow also the dependencies of dependencies etc.
  - Local change may cause higher coupling.
  - Local change may cause higher cohesion.
  - Local change may cause dependencies to become superfluous.

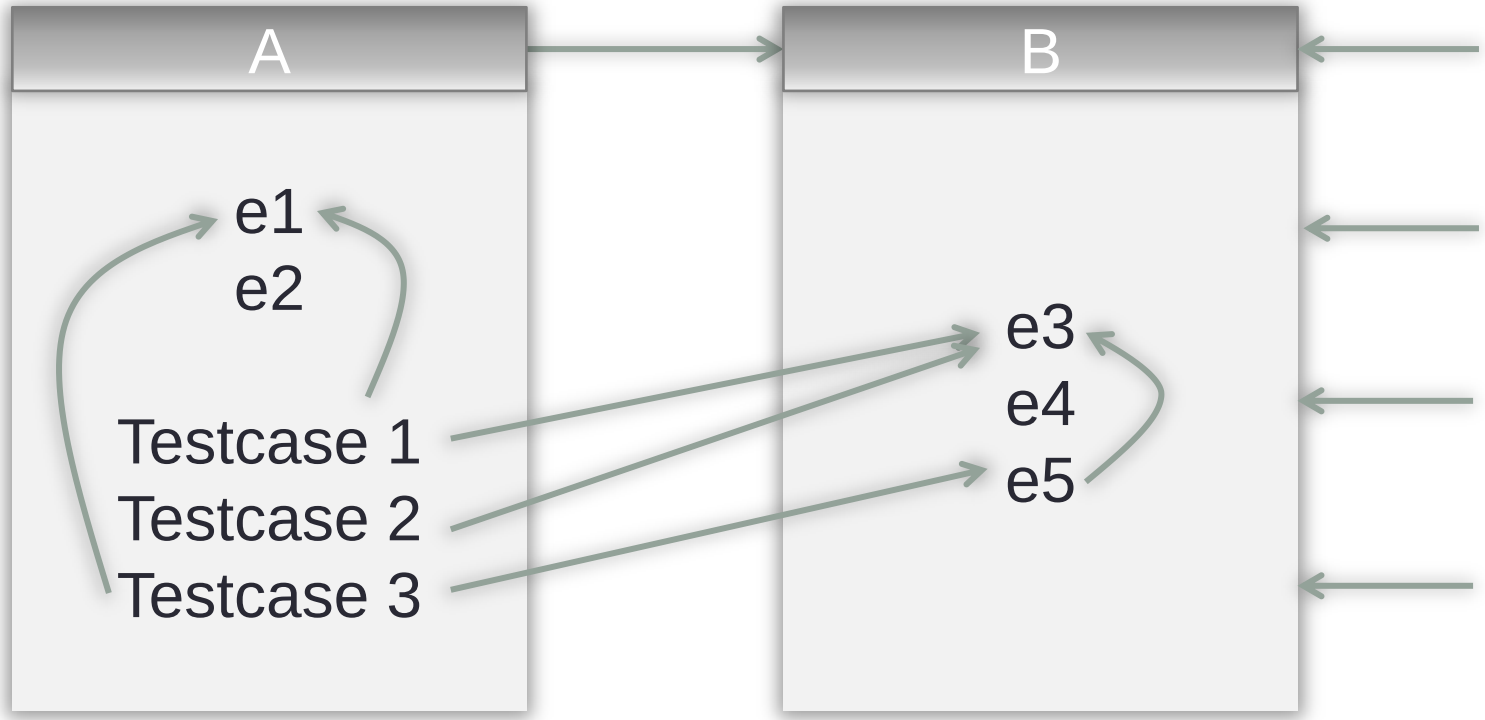
Dependency Analysis

- Element freeze:
  - Testcase 3 is frozen, all dependencies (and dependencies of dependencies) must not be changed anymore



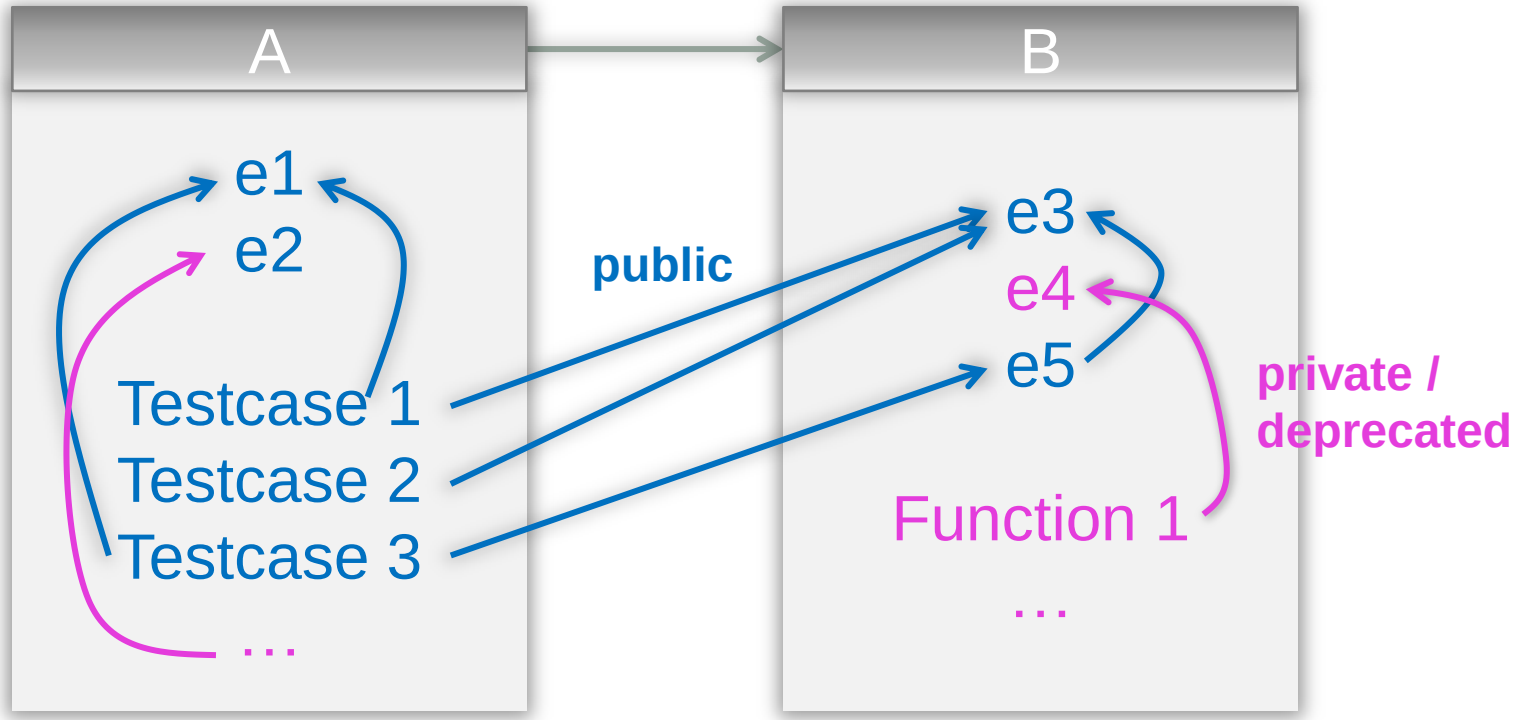
Dependency Analysis

- **Primitive library:**
  - No test cases, no further imports, only incoming dependencies
- **Non-primitive library:**
  - No test cases, *mostly* incoming dependencies



Dependency Analysis

- Public / private / deprecated elements:
  - Note: only useful to kickstart the use of visibility modifiers.



Guideline Analysis

- Guidelines are a ***constructive QA measure*** to prevent mistakes or quality problems.
- Guideline analysis is an ***analytical QA measure*** to continuously enforce guidelines during the development.
- Examples:
  - Naming conventions
  - Test data structuring
  - Style conventions
  - Modularization rules

Guideline Analysis: Naming Conventions

- Examples:
  - Test case numbering:
    - **TC_Syn_0501_Identifier_001**
      (<"TC">"_"<Group index>"_"<TC number>)
  - Non-Default altstep prefix:
    - **a**_receiveSetup()
  - Default altstep prefixes:
    - **d**_receiveSetup()
- Implications:
  - Better understandability

Guideline Analysis: Test Data Structuring

- Examples:
  - Grouping of related definitions
  - Alphabetic ordering of definitions within groups
  - Order and placement of local definitions
- Implications:
  - Improved locality ➔ Better understandability,
    Better maintainability

Guideline Analysis: Style Conventions

- Examples:
  - Formatting style
  - Nesting depth of alt-statements
  - Depth of stacked template modifications
- Implications:
  - Better understandability
  - Better maintainability
  - Better reusability

Guideline Analysis: Modularization Rules

- Examples:
  - Module names imply their content
    - TypesAndValues, Templates, …
  - Standard-Imports must exist
    - LibCommonDefs, …
- Implications:
  - Better locality ➔ Better understandability
  - Bundling of elements that belong together

Dependency / Guideline Relationships

- Dependencies promote quality attributes:
  - Bad quality affects dependent modules
  - High fan-in ➔ impact on quality attributes
  - Determination of modules with high risk regarding change

- Guidelines may involve dependencies:
  - No unused imports
  - Standard imports must exist
  - Over-specific runs on clause

Tools

- T3Q (available now!)
  - Static guideline checking
- T3D (available now!)
  - HTML Documentation Generator (Javadoc-like)
- T3Pendency (available soon!)
  - Test-Suite Dependency Analysis
- Open-Source
  - Eclipse Public License (EPL)
  - Based on the TRex infrastructure
  - TTCN-3 v4.1.1 support
  - Cross-platform
  - Command-line tools, scheduled execution possible

T3Q – TTCN-3 Guideline Checker

- Fine-grained XML configuration with project profiles
- Approx. 30 guideline checks implemented
  - Naming conventions (configurable)
  - Log format must match a regular expression
  - No unused definitions on module level
  - Templates module must contain only template definitions
  - No unused imports
  - No "all" keyword in port type definitions
  - No label or goto statements
  - …
- Code formatting
- Basic size metrics (LOC, No. of test cases,…)

T3D – TTCN-3 Documentation Generator

- XML representation of module definition dependencies
- Generation of different switchable views using XSLT:
  - Main view
    - TTCN-3 listings with cross-links
  - Module parameter view
    - Dependencies between test cases and module parameters
  - Import view
    - Import relationships
  - Documentation as HTML
- Customizable look & feel / documentation branding possible

T3D – Main View

Main View
Module Parameter/Testcase View
Import View

show/hide (toggle) notes | toggle all

Module Index
- SIP_MainModule
- SIP_Steps
- SIP_Templates
- SIP_TypesAndConf

Groups

Module Parameters

Constants

Types

Signatures

Templates

Functions

Altsteps

Testcases
- SIP_CC_OE_CE_TI_001

Index / SIP_MainModule / TestPurposesforCallControl / OriginatingEndpoint / Callestablishment / ValidBehaviour / SIP_CC_OE_CE_V_047

```
testcase SIP_CC_OE_CE_V_047 ( inout CSeq loc_CSeq_s )
     runs on SipComponent
     system SipInterfaces {
       var Request v_ACK_Request;
       var Request v_INVITE_Request;
       var Request v_BYE_Request;
       var boolean body_found;
       initPort ( mtc, system );
       v_Default := activate ( defaultCCOE () );

       if ( PX_HOME_REGISTRATION ) {
         iUTRegistration ()
       };
       action ( "Please send INVITE" );
       TWait.start ( PX_TWAIT );
       alt {
         [] SIPP.receive ( INVITE_Request_r_3 ) -> value v_INVITE_Request {
           TWait.stop;
           body_found := true;
           setHeadersOnReceiptOfInvite ( v_INVITE_Request );

           if ( ispresent ( v_INVITE_Request.msgHeader.contentEncoding ) ) {
             setverdict ( pass )
           } else {
```

T3D v1.0.2
Generated 2010-06-09 - 02:23:45

T3D – Module Parameter View

Index / SIP_MainModule - Module Parameter/Testcase View

Module Parameters (toggle)

Testcases (toggle)

Module Parameters:

Testcases:

SIP_RG_RT_V_001

Module Parameter	Path
PX_UDP	<< initPort << SIP_RG_RT_V_001
PX_IUT_PORT	<< initUDPport << initPort << SIP_RG_RT_V_001
PX_IUT_IPADDR	<< initUDPport << initPort << SIP_RG_RT_V_001
PX_ETS_PORT	<< initUDPport << initPort << SIP_RG_RT_V_001
PX_ETS_IPADDR	<< initUDPport << initPort << SIP_RG_RT_V_001
PX_TWAIT	<< << PX_TWAIT << SIP_RG_RT_V_001

SIP_RG_RT_V_002

Module Parameter	Path
PX_UDP	<< initPort << SIP_RG_RT_V_002
PX_IUT_PORT	<< initUDPport << initPort << SIP_RG_RT_V_002
PX_IUT_IPADDR	<< initUDPport << initPort << SIP_RG_RT_V_002

T3D – Import View

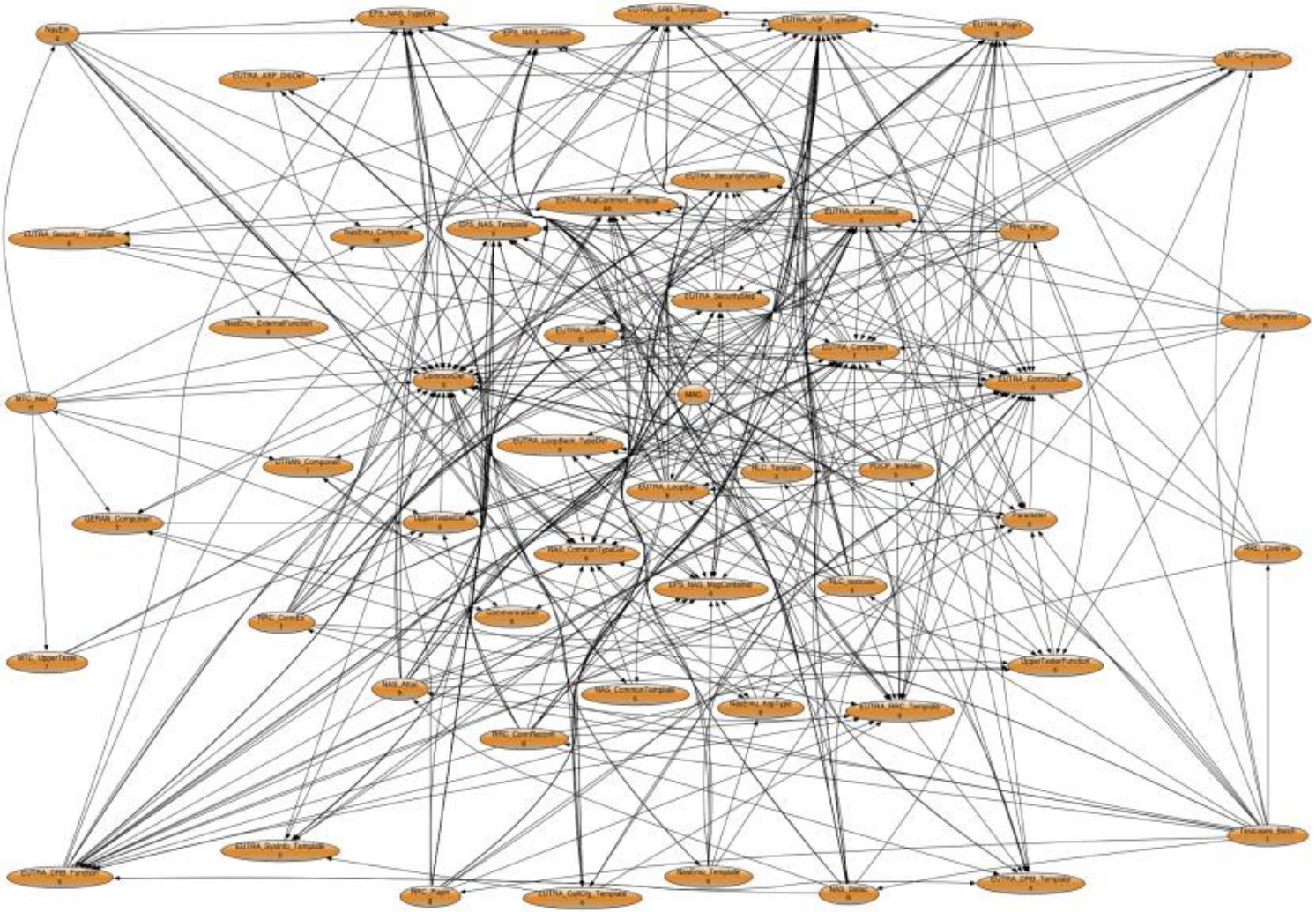
Index / SIP_Templates - Import View

imports	Modules	imported by
SIP_TypesAndConf all	SIP_MainModule SIP_Steps SIP_Templates SIP_TypesAndConf	SIP_MainModule all SIP_Steps all

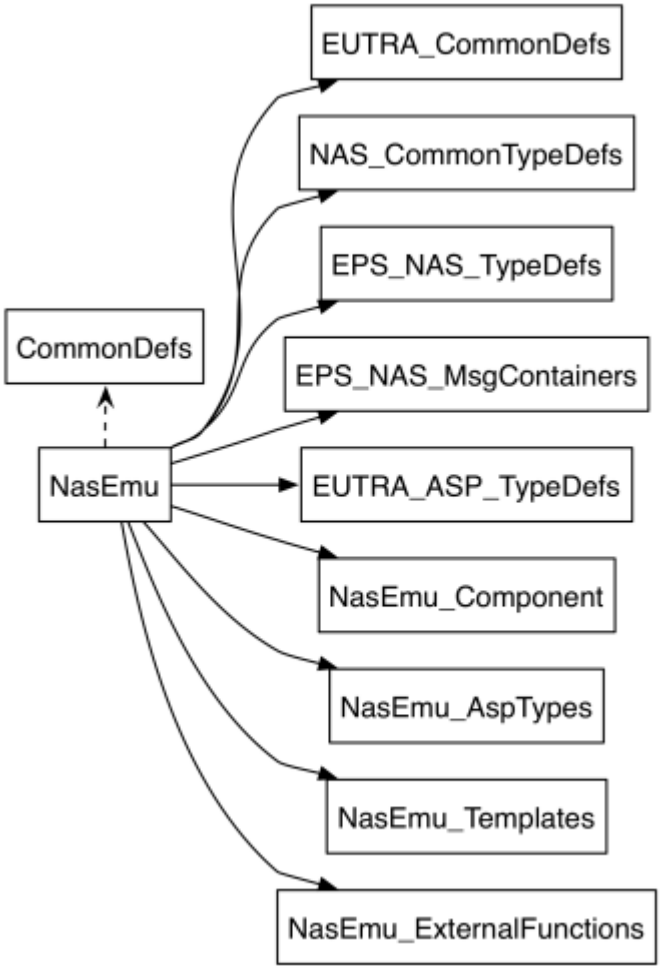
T3Pendency – TTCN-3 Dependency Analyzer

- Calculation of dependency metrics:
  - Number of Imports / Number of superfluous Imports
  - Number of modules that reference a given module (module fan-in)
  - Number of modules referenced by a given module (module fan-out)
  - Number of internal / external references
- Can be determined at the level of:
  - Modules
  - Module definitions
- Public / private suggestions
- Focused Graphviz visualization

T3Pendency – TTCN-3 Dependency Analyzer



T3Pendency – TTCN-3 Dependency Analyzer



Summary & Outlook

- Summary:
  - Dependency analysis
  - Guideline analysis
  - Relationships between Dependencies and Guidelines
  - T3Q, T3D, T3Pendency tools

- Outlook:
  - Freely available, open-source (EPL), T3Pendency soon!
  - Download at http://t3tools.informatik.uni-goettingen.de
  - TRex for Refactoring and Metrics, IDE:
    - http://www.trex.informatik.uni-goettingen.de
  - More guideline checks, more features, but …
    - No commercial support → community-driven tool maintenance!

Contact

- Websites:
  - http://www.trex.informatik.uni-goettingen.de
  - http://t3tools.informatik.uni-goettingen.de
- E-Mail:
  - t3tools@informatik.uni-goettingen.de
- Acknowledgments:
  - ETSI CTI, STF 160