

Papel dos Requisitos Funcionais do Usuário no Ciclo de Engenharia de Subestações Digitais

CIGRE Working Group B5.64

Iony Patriota de Siqueira
ioniy@tecnix.com.br



Contents

Cigre Working Group

Engineering Cycle

Requirement Language

Language Applications

Cigre WG B5.64 Objectives

METHODS FOR SPECIFICATION OF FUNCTIONAL REQUIREMENTS OF PROTECTION, AUTOMATION, AND CONTROL

WG B5.64

Members

Iony P Siqueira, Convenor	BR
Alexander Apostolov	US
Carlos Rodriguez	SP
Paul Myrda	US
Shaoguan Hu	CN
Steven Blair	UK
Volker Leitloff	FR

Luciano Calmon, Secretary	BR
Bruno André	FR
Nirmal-Kumar Nair	NZ
Peng Zhang	US
Simon Hussey	IE
Thomas Charton	UK

Corresponding Members

Adeyemi Adewole	ZA
Daniel Espinosa	MX
Marco Valente	IT
Ratan Das	US
Ronan Keating	IE
Uyttersprot Luc	BE
Zhaogun Meng	CN

Ana Isabel Brito	PT
Marcio Egydio	BR
Nasser Faarooqui	NZ
Rogério Dias Paulo	PT
Timofey Busygin	RU
Yashwant Kodali	IN

- [Survey](#) the current methods and formats for specification of functional requirements of PAC systems
- [Paper](#) in the Cigre Science & Engineering Journal
- [Brochure](#) proposing a Domain-Specific Language (DSL) for specification of functional requirements of PAC solutions
- [Summary](#) in Electra Magazine
- [Tutorial](#) on e-CIGRE

A review of international industry practices for specification of functional requirements of protection, automation and control

Iony PATRIOTA DE SIQUEIRA [A]*, N.U. FAAROOQUI[B], N.K.C. NAIR [B]
On behalf of CIGRE Task Force B5.02

**CIGRE SCIENCE
& ENGINEERING**

^[A]Tecnix Engineering and Architecture Ltd, Brazil

^[B]University of Auckland, New Zealand

- Replies from 135 experts in 97 companies from 42 countries
- Lack of formal methods for specification of functional requirements
- IEC 61850 is the de-facto standard for designing PAC systems
- Any new PAC Requirement Language should be:
integrated to IEC 61850, readable by humans and computers, understandable by non-expert users, formal and compilable to design languages

Engineering Design Cycle

Workshop

Engineering Design Cycle

Waterfall Engineering Cycle

V-Model Engineering Cycle

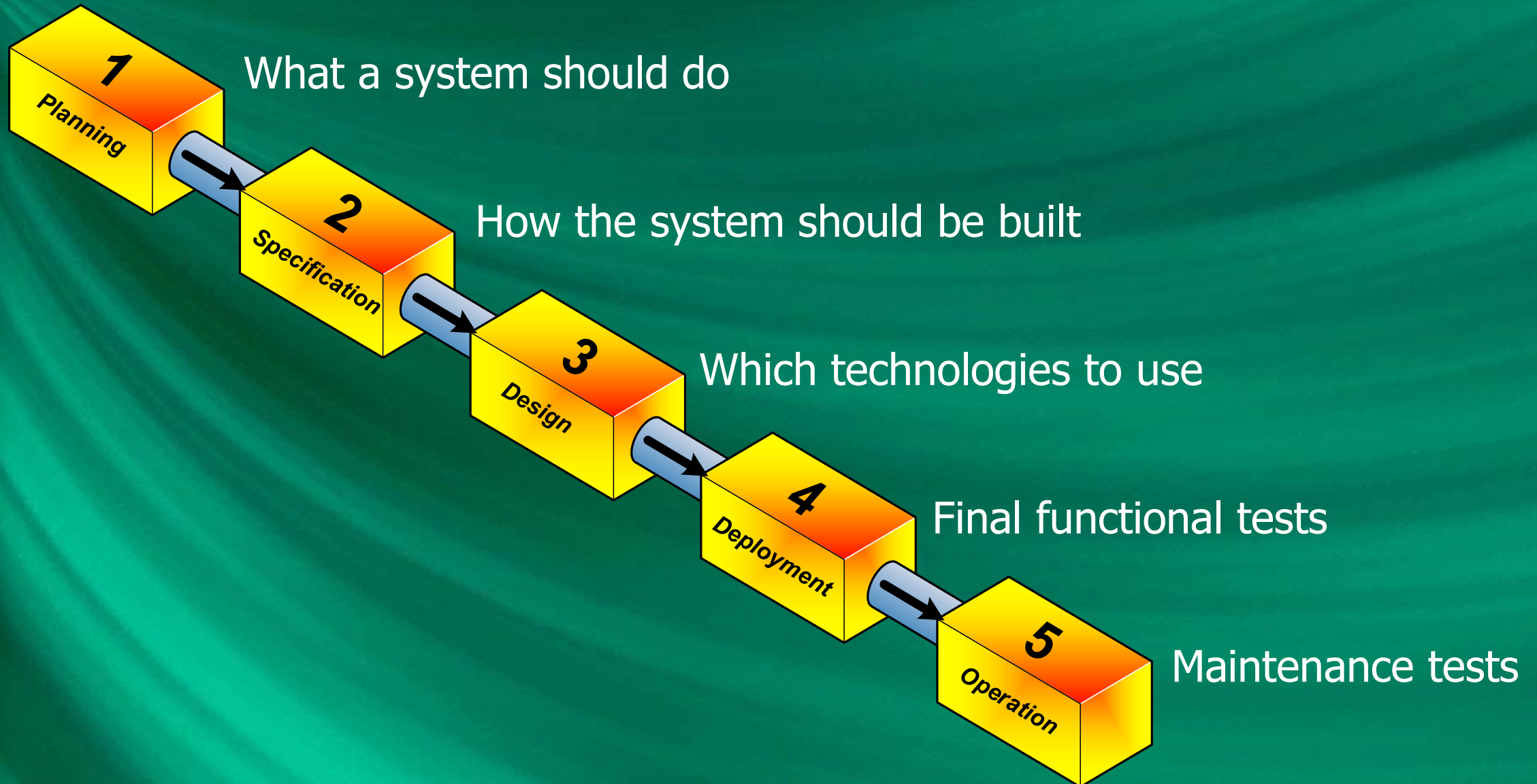
Agile Engineering Cycle

Engineering Specifications

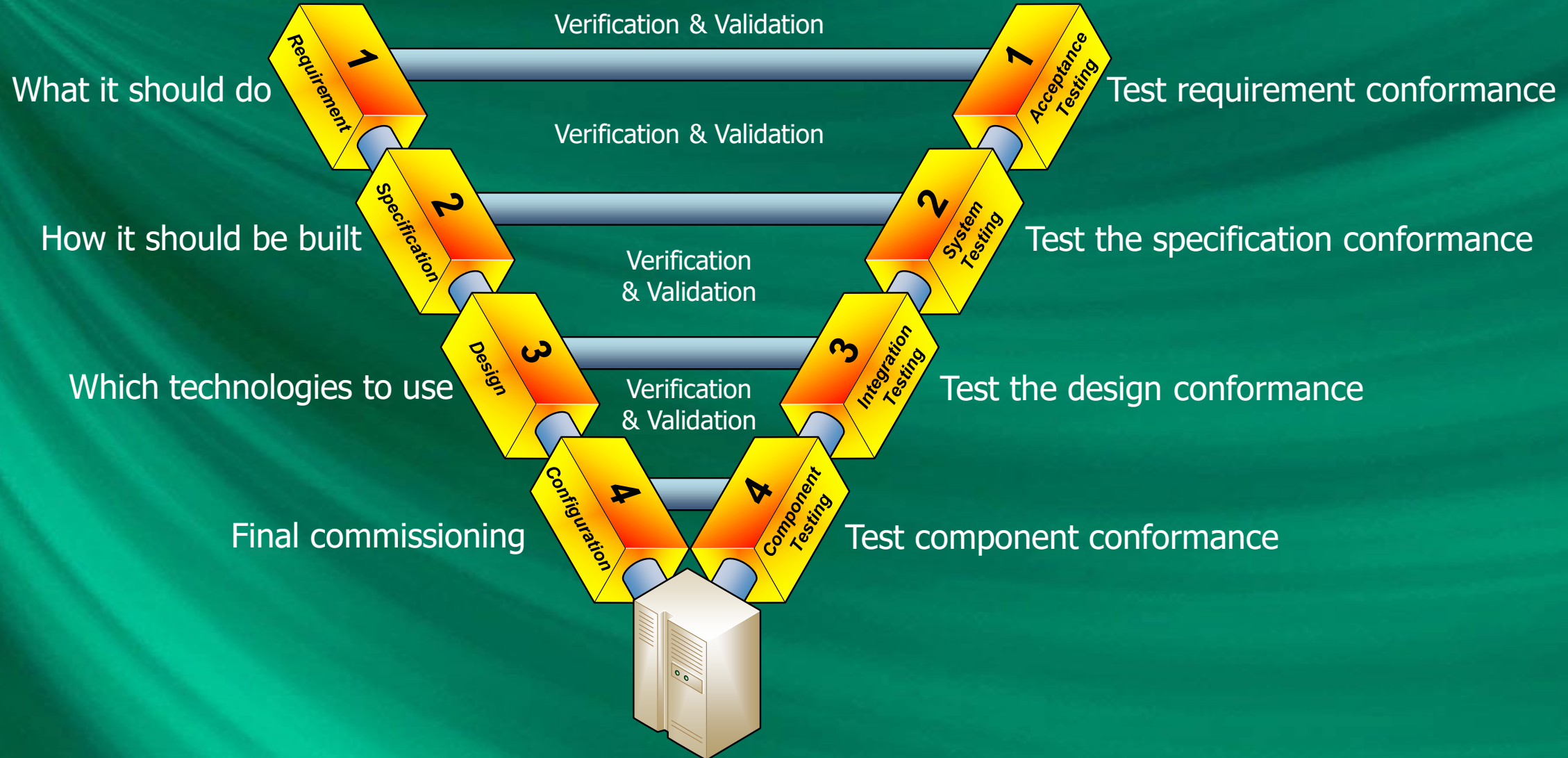
PAC Engineering Process

Waterfall Engineering Process

Workshop

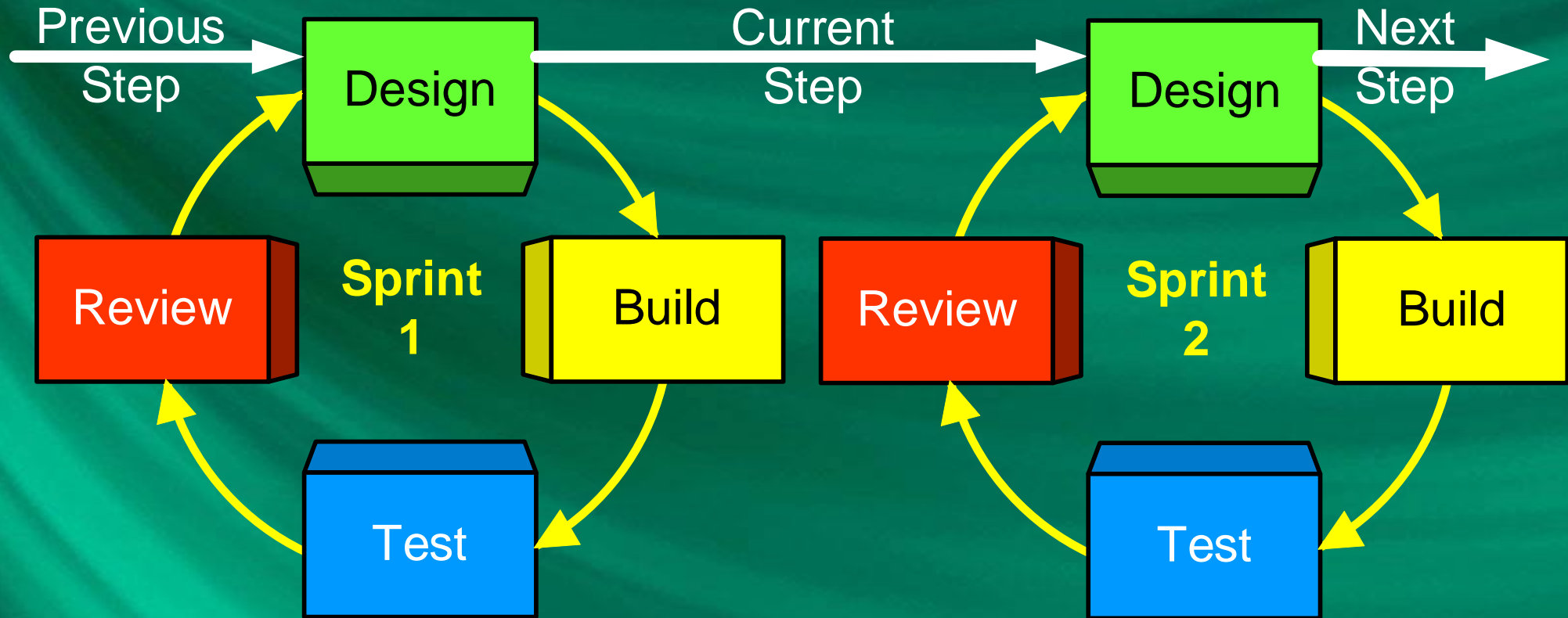


V-Model Engineering Cycle



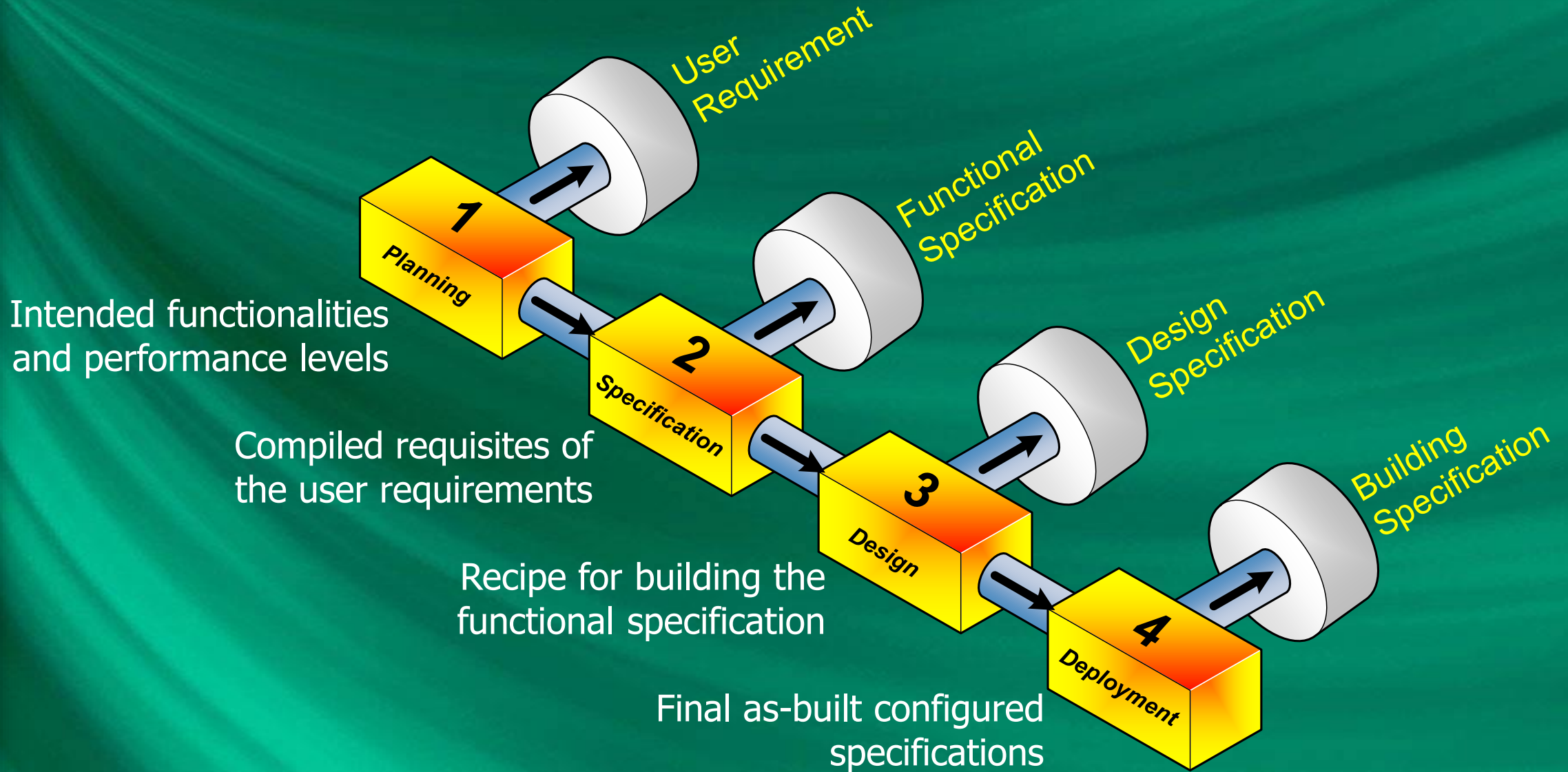
Agile Engineering Cycle

Workshop



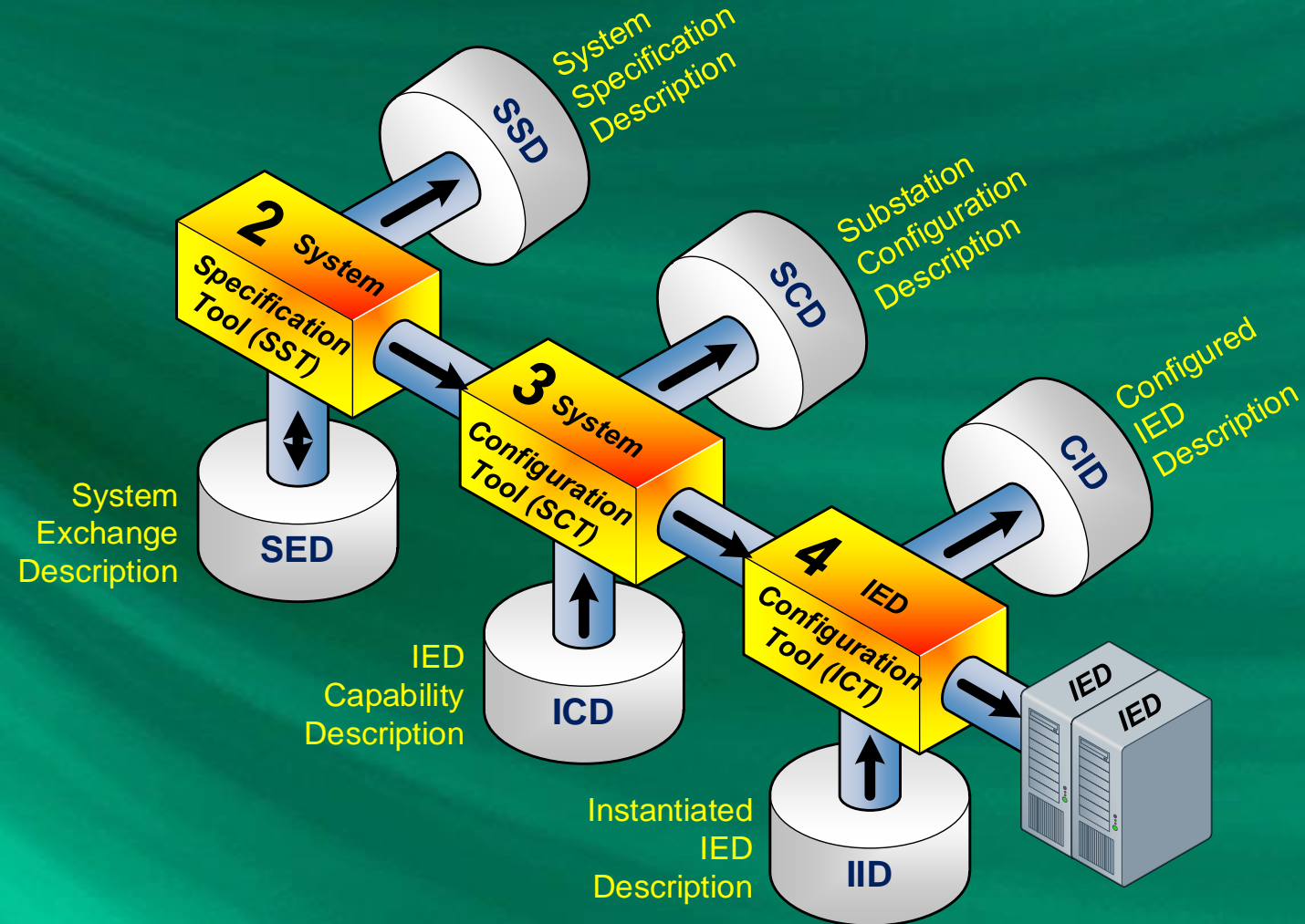
Engineering Specifications

Workshop

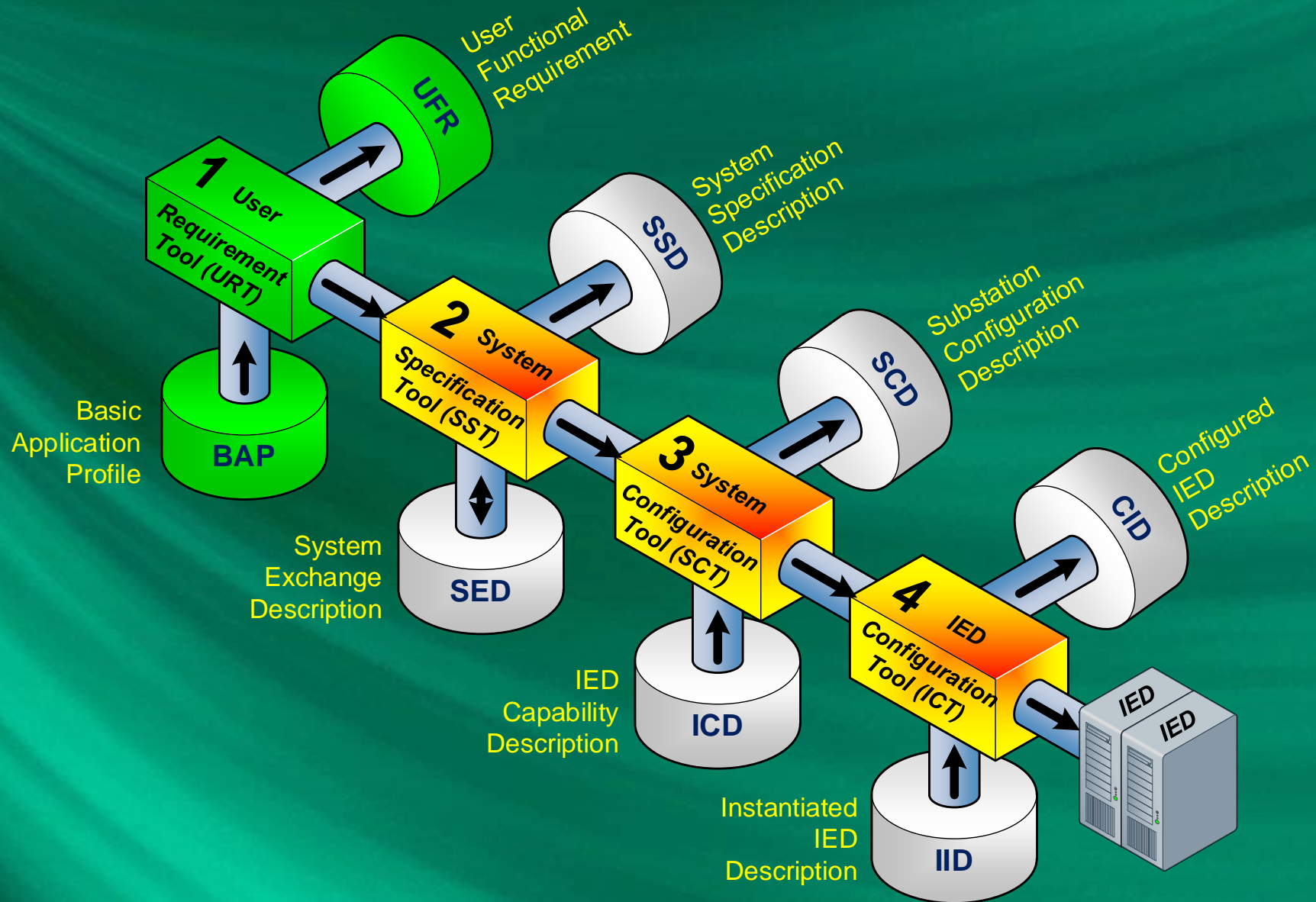


IEC 61850 Engineering Process and Tools

Workshop



IEC 61850 Engineering Cycle with User Requirement



Requirement **Language** Definition

Workshop

Domain Specific Language

Lexicon Definition

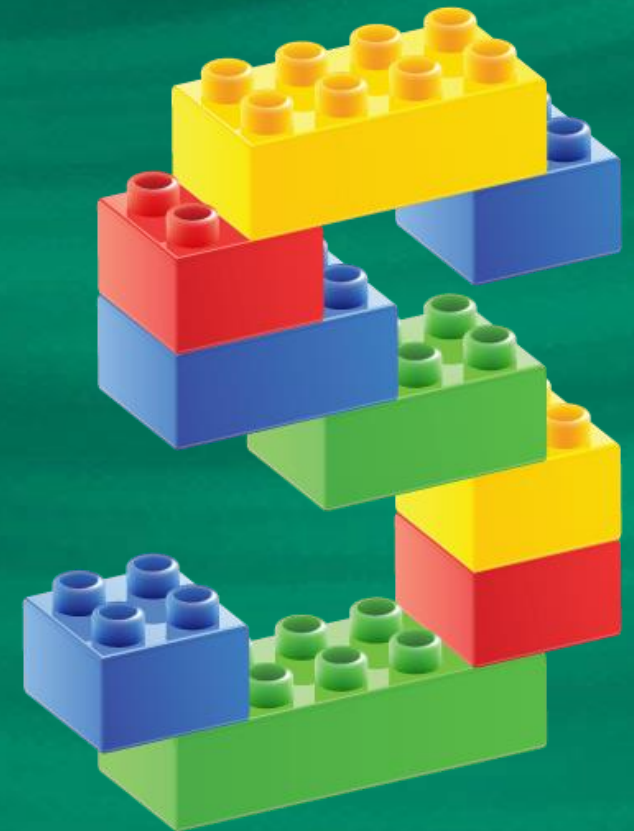
Syntax Definition

Semantics Definition

Pragmatics Definition

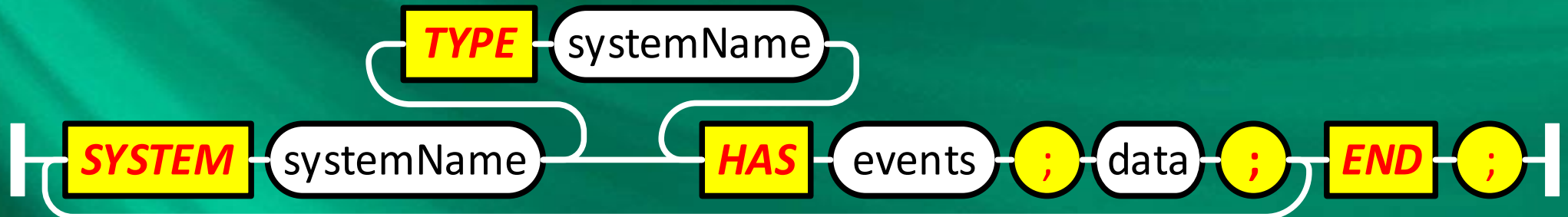
Lexicon Definition

- Definition
 - The vocabulary or symbols (lexemes) used by a language
- Components
 - Characters: A ... Z, a ... z, 0 ... 9
 - Delimiters: ()[]{}.,;/*'"
 - Vocabulary: sequence of characters
 - Numbers: sequence of digits
 - Keywords: with predefined meaning
 - Operators: > < ≥ ≤ = ≠ + − × ÷
 - Comments: /* ... */ or //

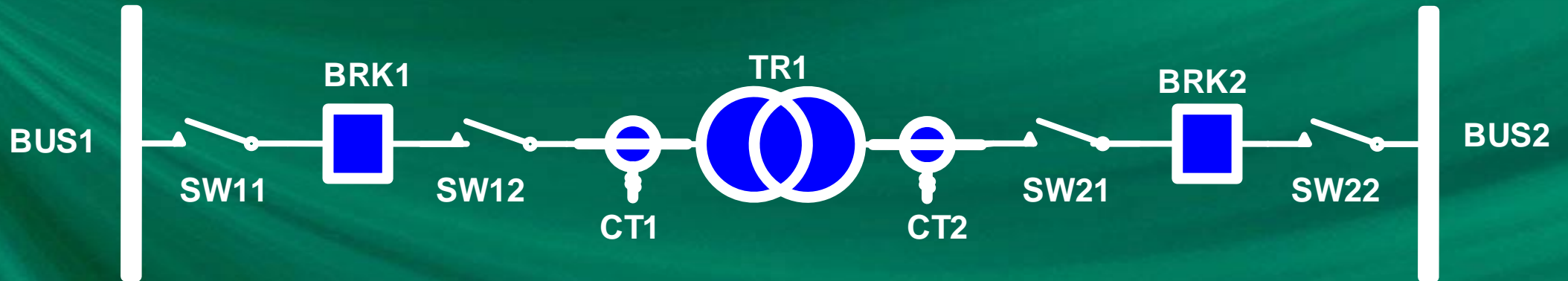


Syntax Definition

- Definition
 - Set of rules to combine the symbols in well-formed phrases
- Components
 - systems, messages, data, timing, temporal logic
- Metalanguage
 - BNF: Backus-Naur Form



Syntax Example



SYSTEM Bay **HAS**

SYSTEM Transformer1 **HAS**

EVENT Internal-Short-Circuit-to-Ground;

EVENT Internal-Phase-Short-Circuit;

EVENT Inter-Turn-Coil-Short-Circuit;

EVENT Bushing-Ground-Short-Circuit;

EVENT Bushing-Phase-Short-Circuit;

DATA Phase-Current;

DATA Phase-Voltage;

DATA Oil-Temperature;

DATA Capacity;

END; /* system Transformer1 */

SYSTEM Breaker1 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

DATA Gas-Pressure;

END; /* system Breaker1 */

SYSTEM Breaker2 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

DATA Gas-Pressure;

END; /* system Breaker2 */

SYSTEM CT1 **HAS**

DATA Current;

END; /* system CT1 */

SYSTEM SW11 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

END; /* system SW11 */

SYSTEM SW12 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

END; /* system SW12 */

SYSTEM SW21 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

END; /* system SW21 */

SYSTEM SW22 **HAS**

EVENT Opening;

EVENT Closing;

DATA Position;

END; /* system SW22 */

SYSTEM CT1 **HAS**

DATA Current;

END; /* system CT1 */

END; /* system Bay */

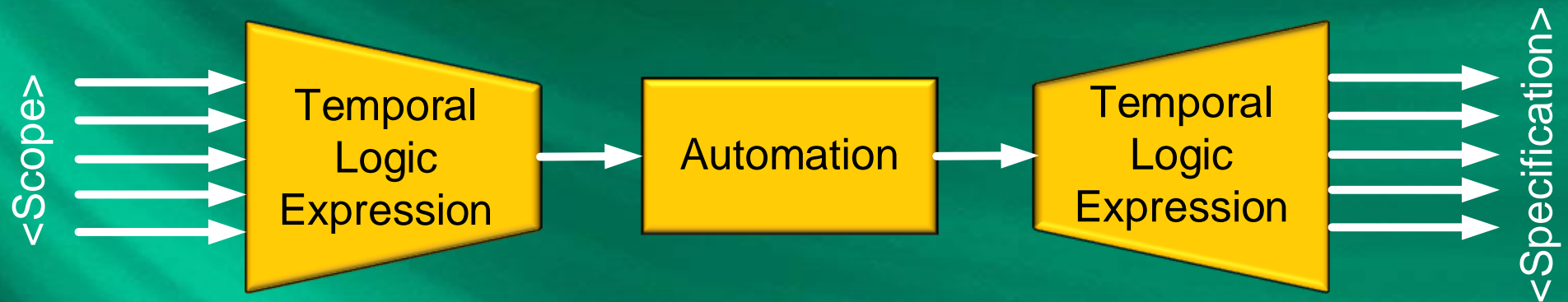
Semantics Definition

- Definition

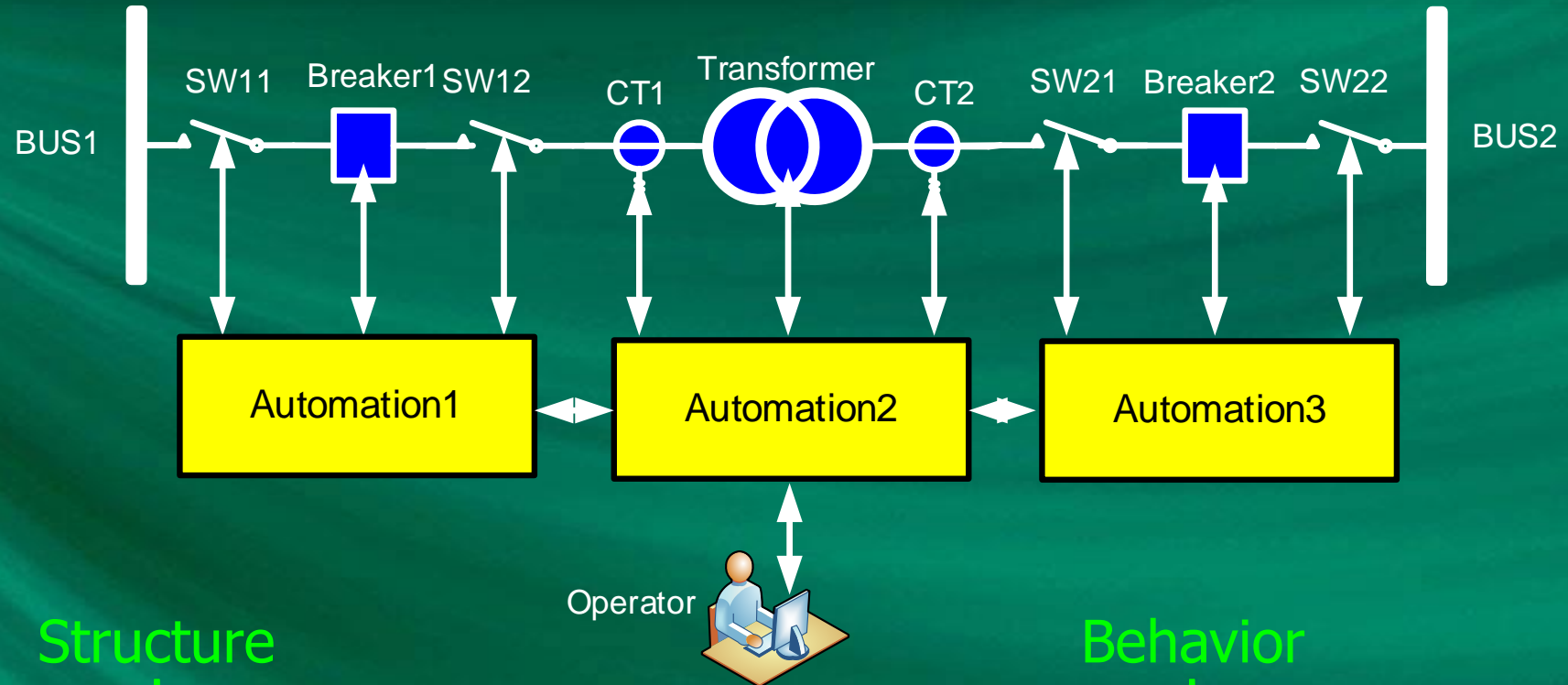
- Set of rules to specify meaning and behavior to any well-formed construct or phrase

- Format

- Informal – natural language description (like IEC 61850)
- Structural – hierarchy and inheritance of organizational elements
- Axiomatic – dynamics of events and data transformation



Semantics Example



Structure

Behavior

SYSTEM Automation1 **HAS**
EVENT Trip;
EVENT Close;
EVENT Alarm;
END;
SYSTEM Automation2 **HAS**
EVENT Trip;
EVENT Close;
EVENT Alarm;
END;

SYSTEM Automation3 **HAS**
EVENT Trip;
EVENT Close;
EVENT Alarm;
END;
SYSTEM Operator **HAS**
EVENT Open;
EVENT Close;
END;

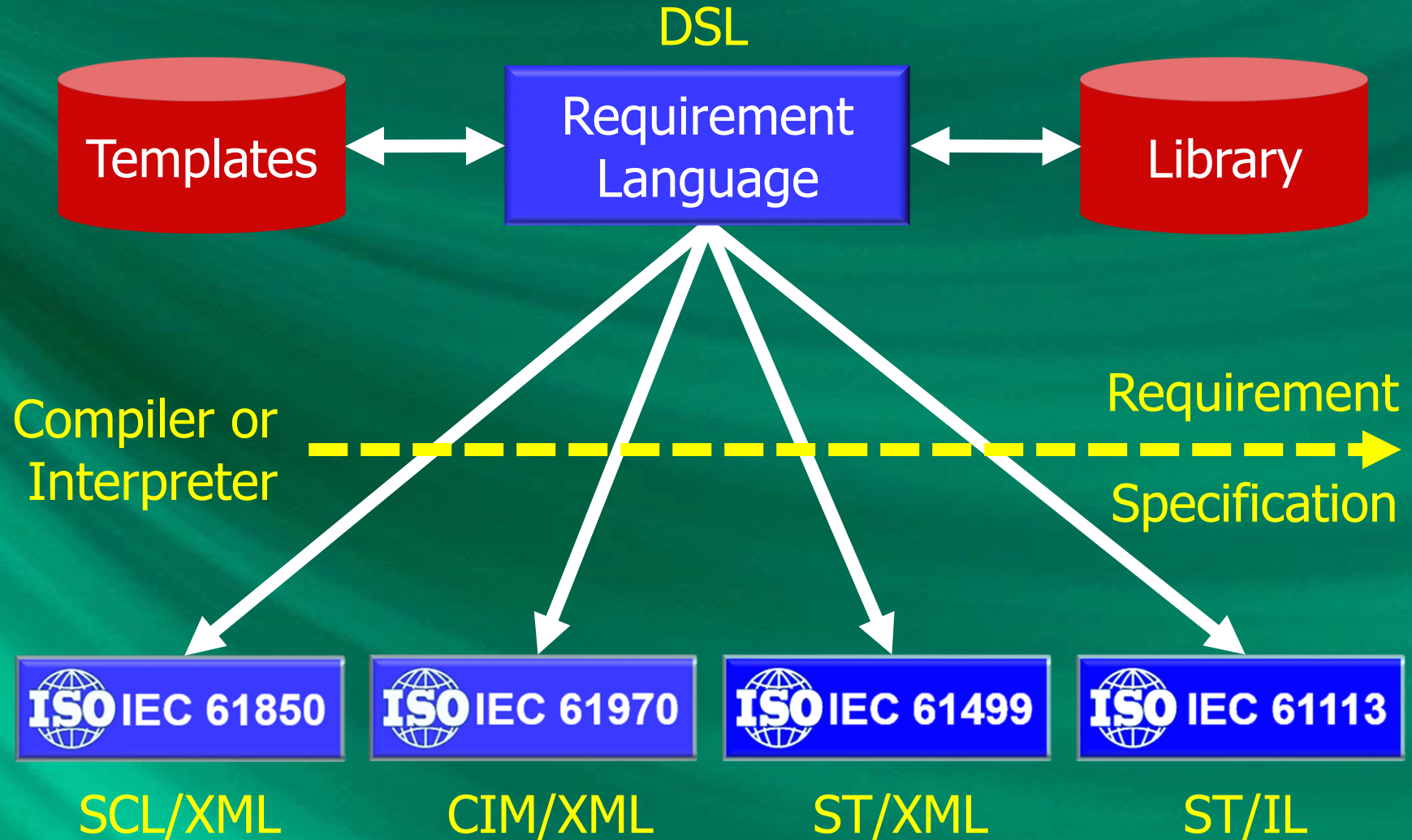
AFTER Transformer.Internal-Short-Circuit **THEN**
Automation1.Trip **TO** Breaker1 **MUST OCCUR BEFORE A DELAY OF** 100 ms,
Automation2.Trip **TO** Breaker2 **MUST OCCUR BEFORE A DELAY OF** 100 ms;
AFTER Transformer.External-Short-Circuit **THEN**
Automation1.Trip **TO** Breaker1 **CANNOT OCCUR BETWEEN A DELAY OF** 0 ms
AND A DELAY OF 100 ms;
AFTER CT1.Current > Transformer1.Capacity **THEN**
Automation1.Trip **TO** Breaker1 **AFTER A DELAY OF** 100 ms,
Automation2.Trip **TO** Breaker2 **AFTER A DELAY OF** 100 ms;

Pragmatics Definition

- Definition
 - Set of available methods and tools for the practical use of the language
- Components
 - DSL Development Environments
 - Requirement Development Environments
 - Interpreters and Compilers
 - Standard Library Modules
 - Application Profiles & Templates



Compilers and Interpreters



XText Language Development Environment

Workshop

Language
Definition

Language
Structure

Language
Package

Syntax
Diagram

PacDsl - org.xtext.example.PacDsl/src/org.xtext.example.mydsl/PacDsl.xtext - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Pack... PacDsl.xtext IEC61850.pacdsl

```
17 Event: "Event" name=Idorstring (array ?= '[' (Length=INT)? '']'? ('='
18 Data: "Data" name=Idorstring (array ?= '[' (Length=INT)? '']'? ('=' Te
19 EventReference : EventRefer | EventRef;
20 DataReference : DataRefer | DataRef;
21 EventRefer : {EventRefer} name=[Event|Idorstring] (array ?= '[' (Lengt
22 DataRefer : {DataRefer} name=[Data|Idorstring] (array ?= '[' (Length=I
23 EventRef : {EventRef} name=DotExpression ('to' Destination+=DotExpress
24 DataRef : {DataRef} name=DotExpression;
25 Fact: DataReference ('='| '<'| '>'| '<='| '>='| '<=') Text;
26 ... returns ecore::ENamedElement : {Requisites} ((NameRequisite+
27 Requisite: 'Requisite' name=Idorstring 'needs' ('|')? NameScope=Scope
28 RequisiteReference : 'Requisite' name=RequisiteRef;
29 Scope returns ecore::ENamedElement : {Input} (NameScopebody=Scopebody);
30 Scopebody : {Scopebody}
31   ( 'always' | 'never' | 'eventually' | 'at' 'most' Integer 'tim
32     | 'while' ConditionWhile+=Condition
33     | 'before' ConditionBefore+=Condition
34     | 'after' ConditionAfter+=Condition (('prior' 'to' ConditionPr
35     | 'between' ConditionBetween+=Condition 'until' ConditionBetwe
```

Quick Access Outline

- Precede
- Exactly
- After
- Before
- Between
- Repetition
- Occurrence
- Last
- Delay
- Number
- Probability
- DotExpression
- EntityRef
- RequisiteRef
- QualifiedName

Problems Javadoc Declaration Console Xtext Syntax Graph

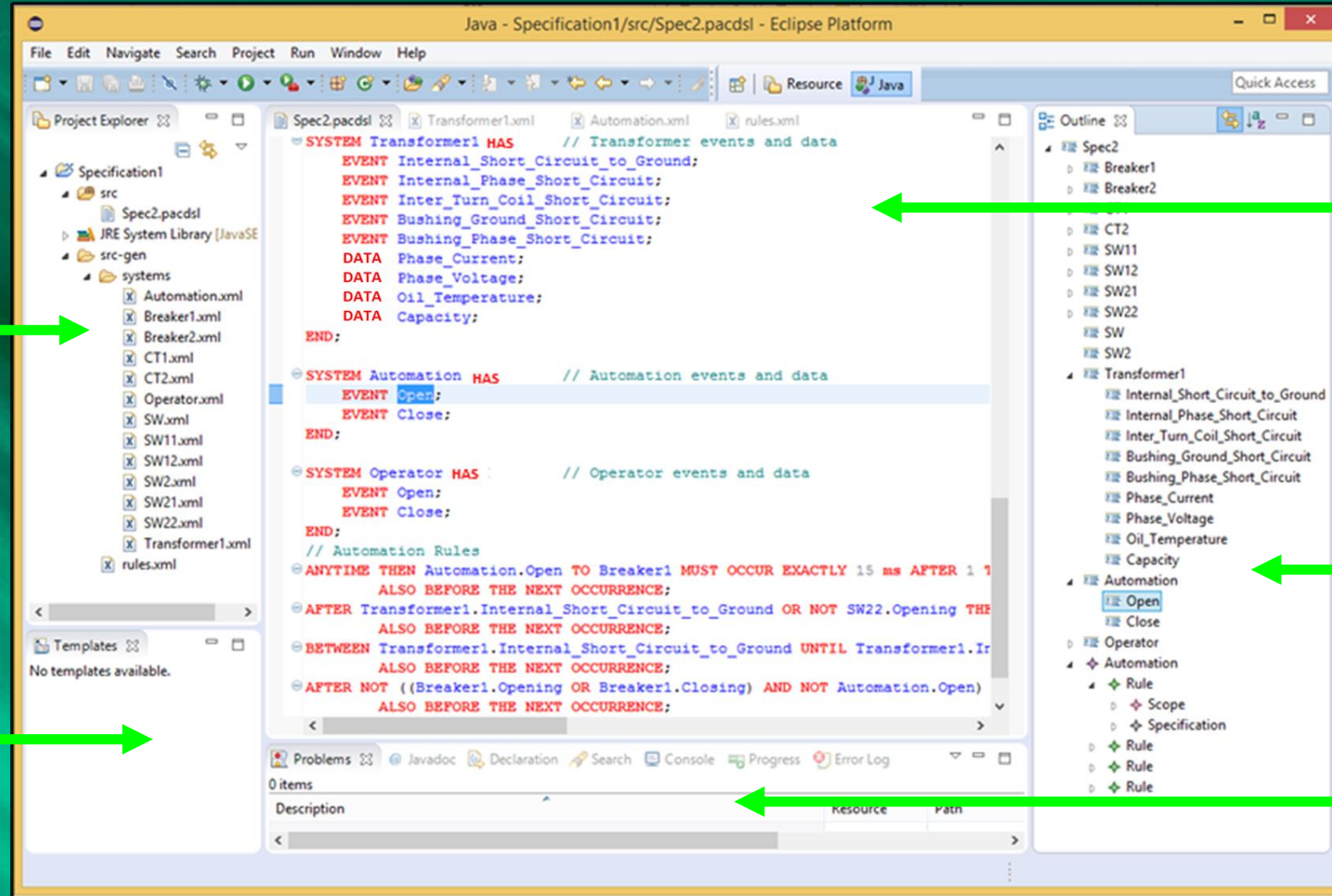
Event - Event - Idorstring - [[INT]] - [=] - Condition

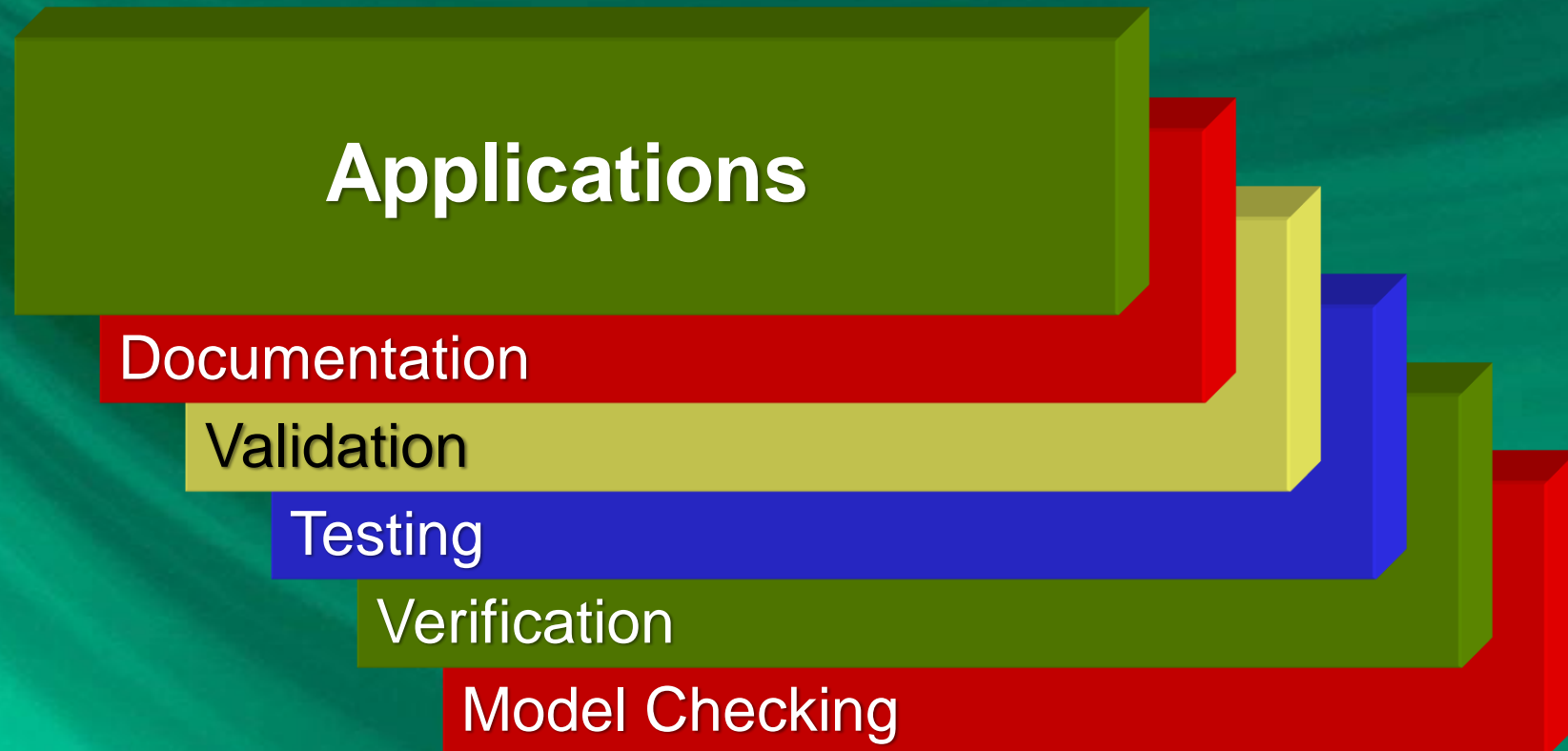
Data - Data - Idorstring - [[INT]] - [=] - Text

322M of 586M

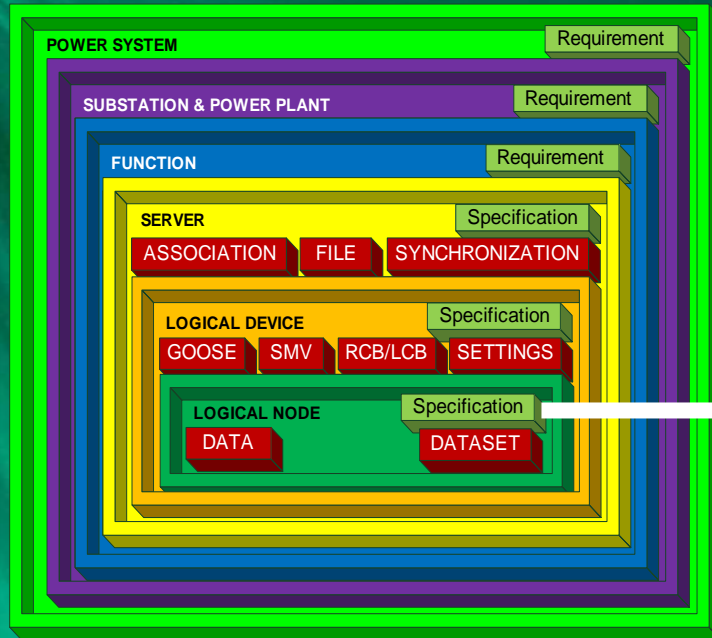
DSL Requirement Development Environment

Workshop





Documentation



SYSTEM PIOC **TYPE** CmmnLN **HAS**

SYSTEM Str **TYPE** ACD **HAS END**; // Start : Optional

SYSTEM Op **TYPE** ACT **HAS END**; // Operate

SYSTEM OpCntRs **TYPE** INC **HAS END**; // Resettable operation counter

SYSTEM StrVal **TYPE** ASG **HAS END**; // Start value : Optional

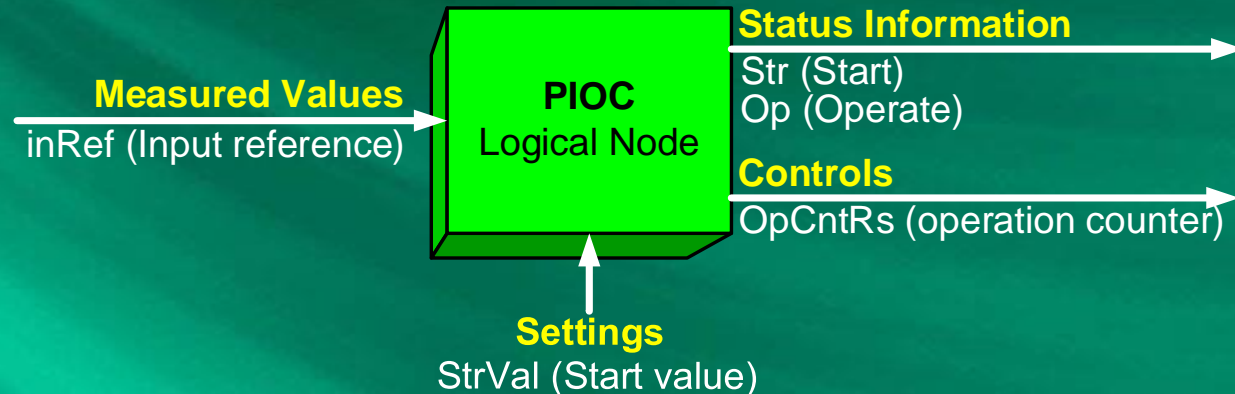
WHEN Str.phsA **IS TRUE** **AND** Str.dirPhsA **IS** "forward" **THEN** Op.phsA **IS TRUE**;

WHEN Str.phsb **IS TRUE** **AND** Str.dirphsb **IS** "forward" **THEN** Op.phsb **IS TRUE**;

WHEN Str.phsc **IS TRUE** **AND** Str.dirphsc **IS** "forward" **THEN** Op.phsc **IS TRUE**;

WHEN InRef1.setSrcRef **IS TRUE** **THEN** Op.general **IS TRUE**;

END;



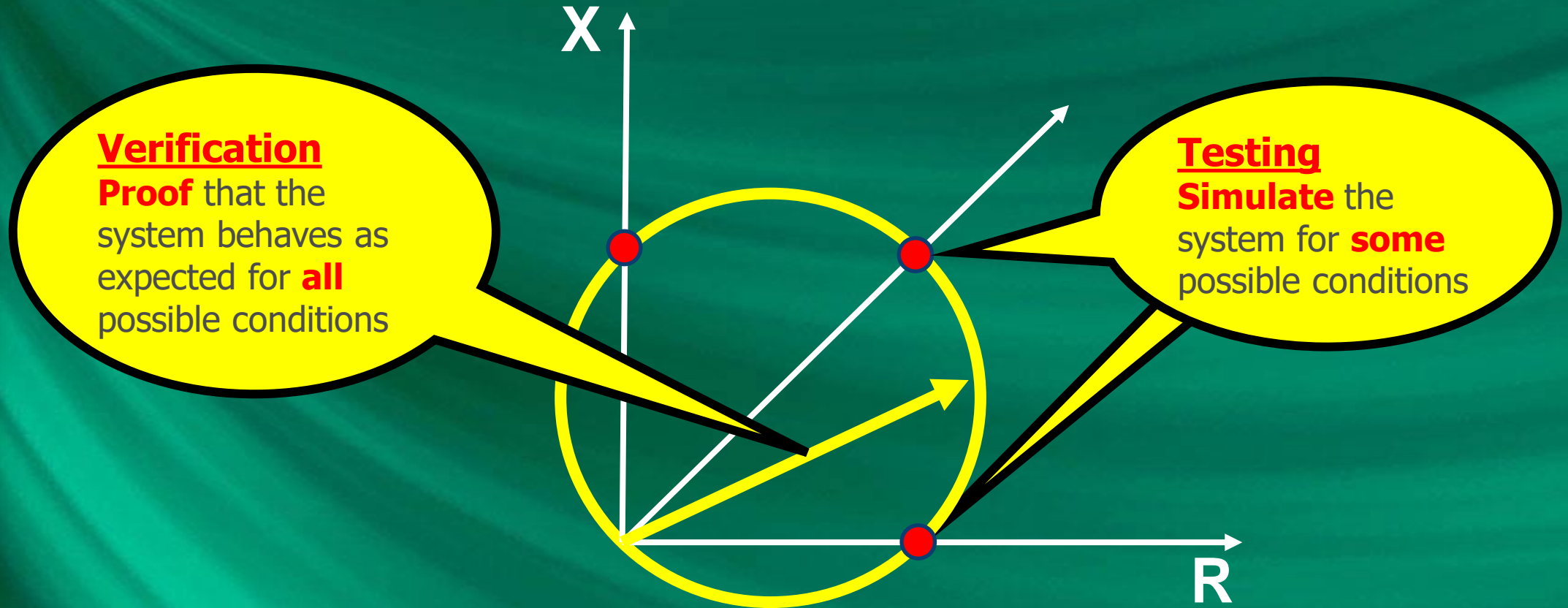
Verification
(Proofing)

Checking that the system
behaves as expected for
all conditions, before
implementation

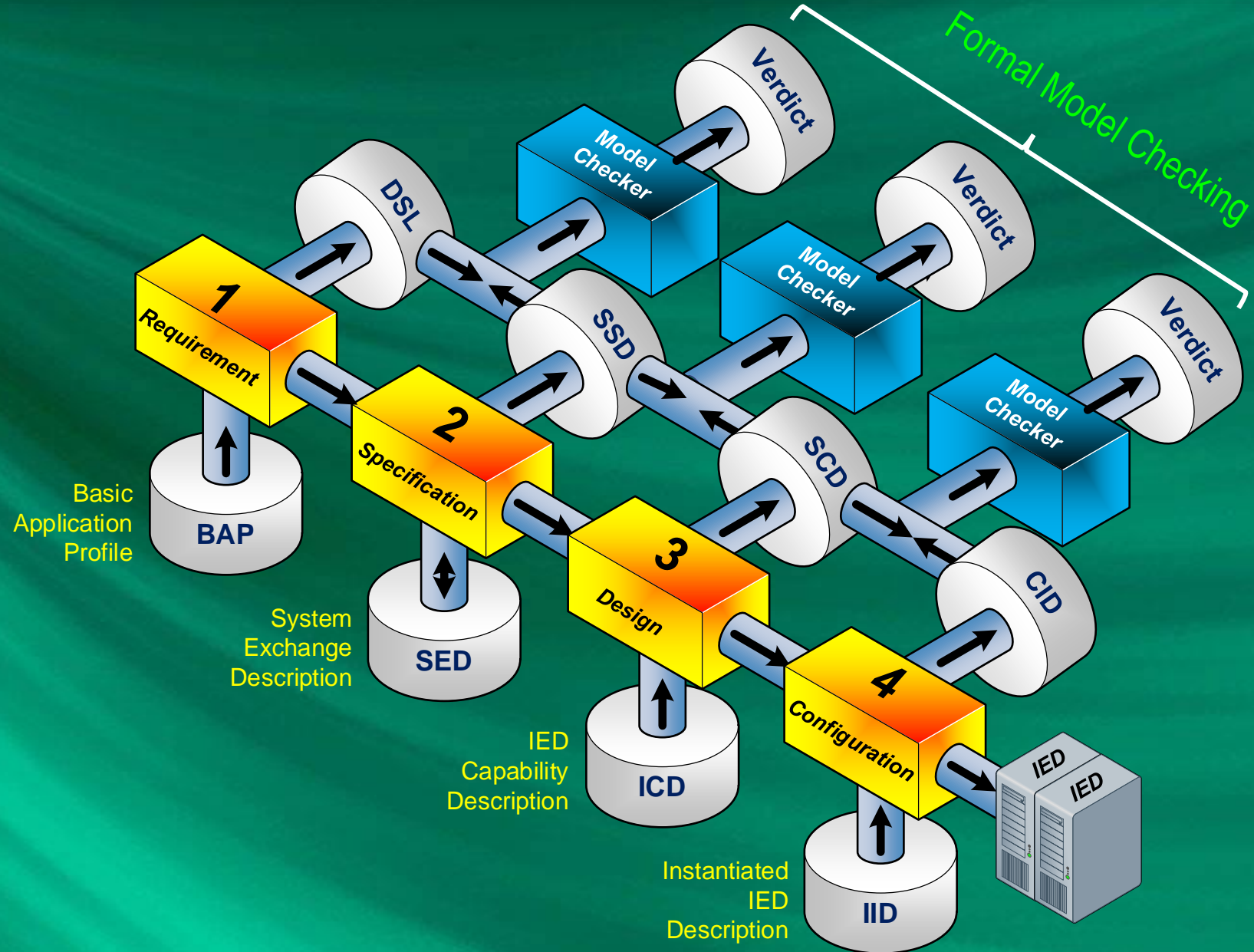
Testing
(Simulation)

Checking that the system
behaves as expected for
some conditions, after
implementation

Verification / Testing



IEC 61850 Model Checking



Obrigado!

CIGRE Working Group B5.64

Iony Patriota de Siqueira
ioniy@**tecnix**.com.br

