Business rules and BREX in a real project

21st of June 2022, Renton

Joakim Lundqvist

Joakim.lundqvist@saabgroup.com

Agenda

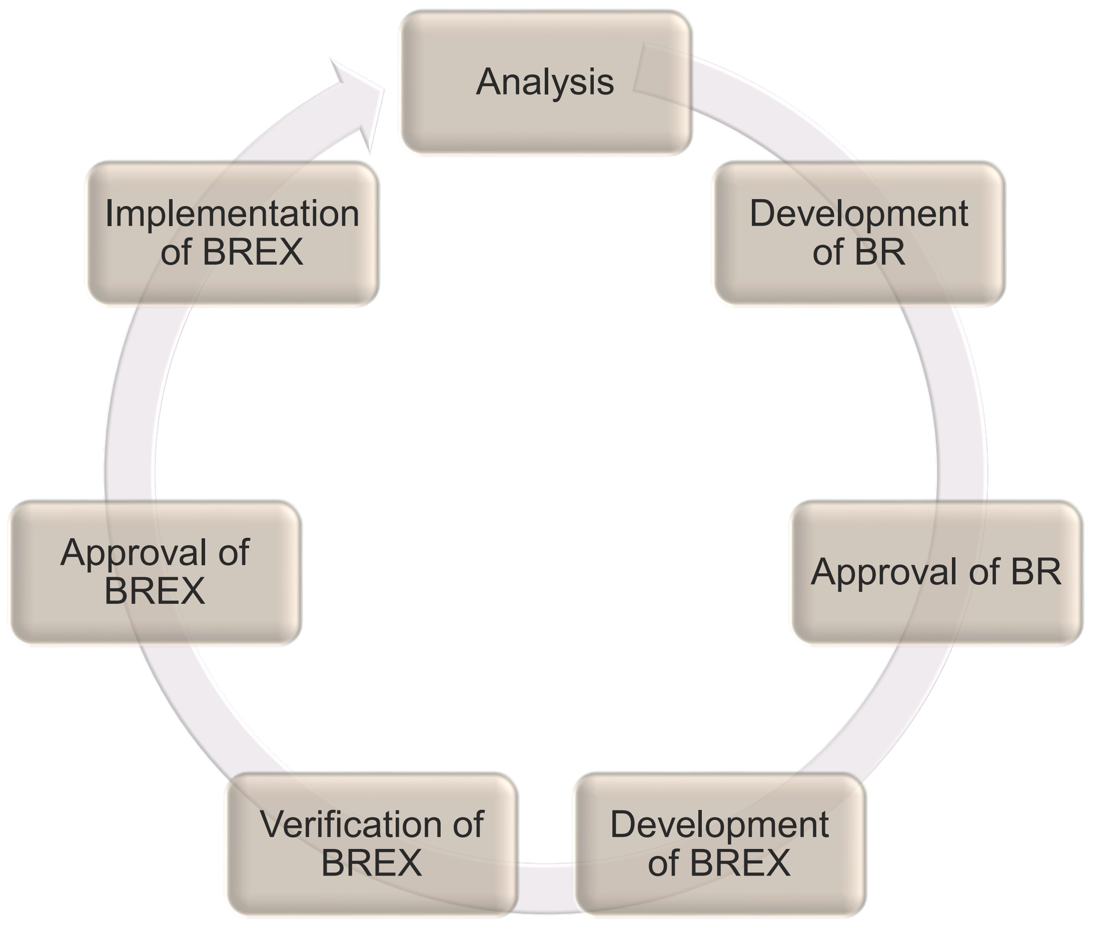
- Project assumptions
- Process
- Development BREX data modules
- Creation of test cases
- Why use a BREX data module?
- BREX checker
- Demo
- Lessons learned

Project assumptions

- S1000D Issue 4.1
- Production System (are in use)
- Layered BREX
- Business rules are agreed

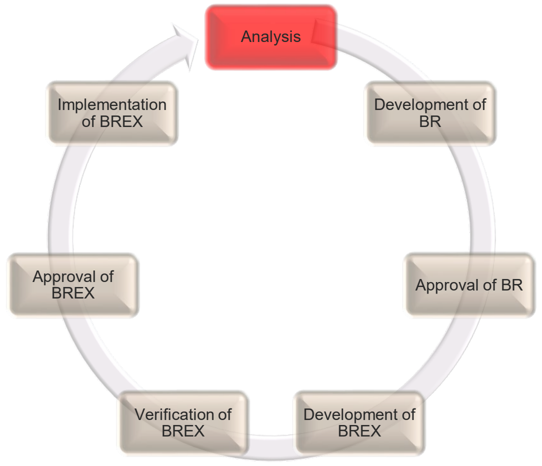
- Example for this presentation is the “Random list prefix”

BREX development process



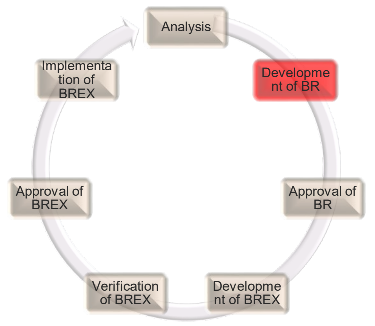
Analysis

- Search and locate affected chapters in the specification
- Read and understand the specification
- Understand the functions in the production system and limitations
- Discuss the project “actual” requirements
  - Include suppliers and partners
  - Include cost
- Discuss and decide (if possible) the desired output result (hard copy/IETP/communication with other DB)



Development of BR

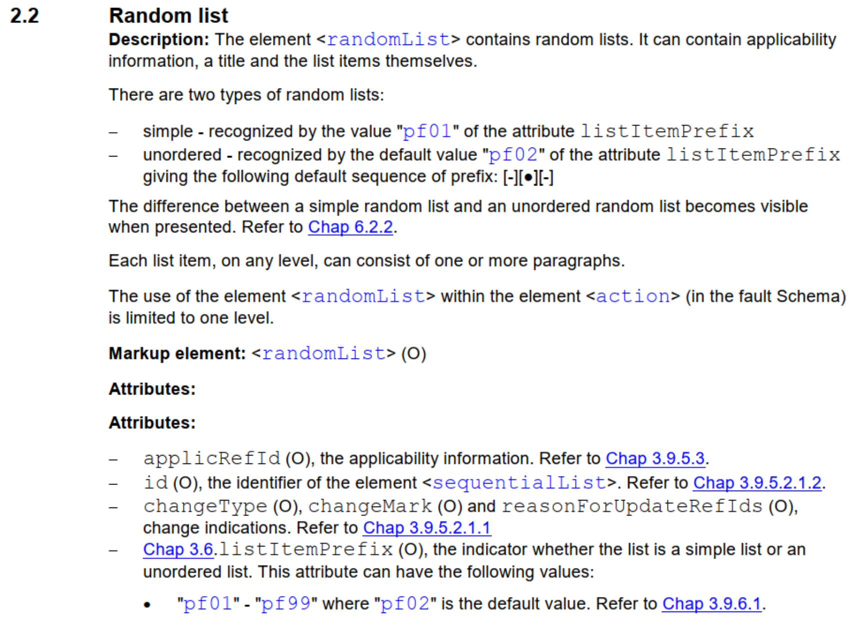
- Business rules decision points
- Other decisions



Chap 3.9.5.1.3 Common construct - Lists

Business rules decisions points (to take):

- define simple and unordered lists and their usage
- use of the attribute listItemPrefix



2.2 Random list

Description: The element <randomList> contains random lists. It can contain applicability information, a title and the list items themselves.

There are two types of random lists:

- simple - recognized by the value "pf01" of the attribute listItemPrefix
- unordered - recognized by the default value "pf02" of the attribute listItemPrefix giving the following default sequence of prefix: [-][•][-]

The difference between a simple random list and an unordered random list becomes visible when presented. Refer to Chap 6.2.2.

Each list item, on any level, can consist of one or more paragraphs.

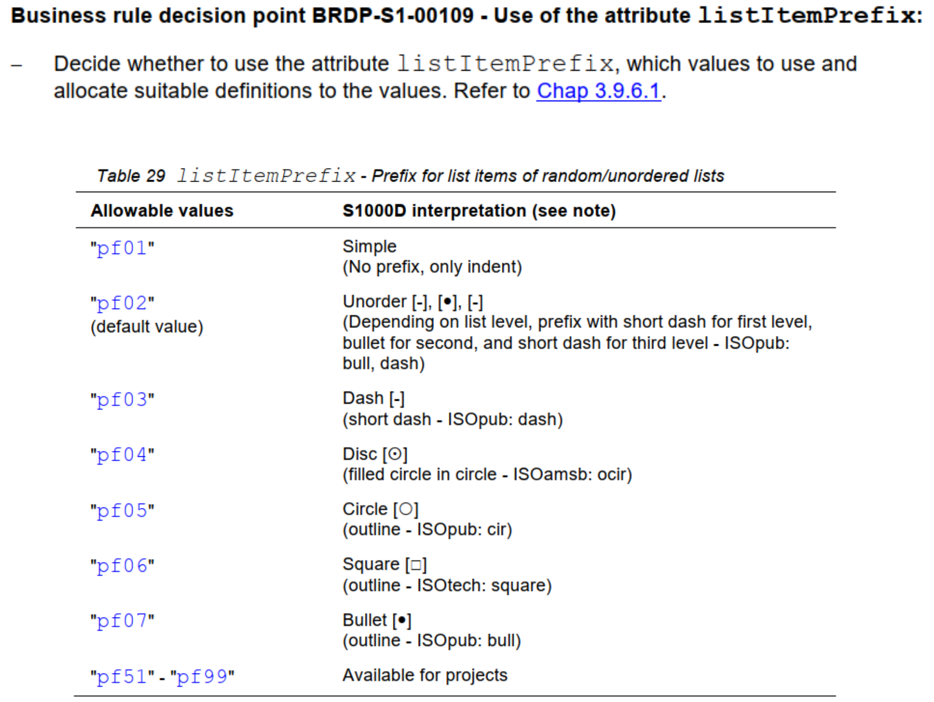
The use of the element <randomList> within the element <action> (in the fault Schema) is limited to one level.

Markup element: <randomList> (O)

Attributes:

Attributes:

- applicRefId (O), the applicability information. Refer to Chap 3.9.5.3.
- id (O), the identifier of the element <sequentialList>. Refer to Chap 3.9.5.2.1.2.
- changeType (O), changeMark (O) and reasonForUpdateRefIds (O), change indications. Refer to Chap 3.9.5.2.1.1
- Chap 3.6.listItemPrefix (O), the indicator whether the list is a simple list or an unordered list. This attribute can have the following values:
  - "pf01" - "pf99" where "pf02" is the default value. Refer to Chap 3.9.6.1.

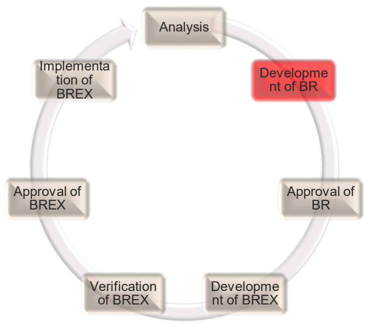


Business rule decision point BRDP-S1-00109 - Use of the attribute listItemPrefix:

- Decide whether to use the attribute listItemPrefix, which values to use and allocate suitable definitions to the values. Refer to Chap 3.9.6.1.

Table 29 listItemPrefix - Prefix for list items of random/unordered lists

Allowable values	S1000D interpretation (see note)
"pf01"	Simple (No prefix, only indent)
"pf02" (default value)	Unorder [-], [•], [-] (Depending on list level, prefix with short dash for first level, bullet for second, and short dash for third level - ISOpub: bull, dash)
"pf03"	Dash [-] (short dash - ISOpub: dash)
"pf04"	Disc [⊙] (filled circle in circle - ISOamsb: ocir)
"pf05"	Circle [○] (outline - ISOpub: cir)
"pf06"	Square [□] (outline - ISOtech: square)
"pf07"	Bullet [•] (outline - ISOpub: bull)
"pf51" - "pf99"	Available for projects

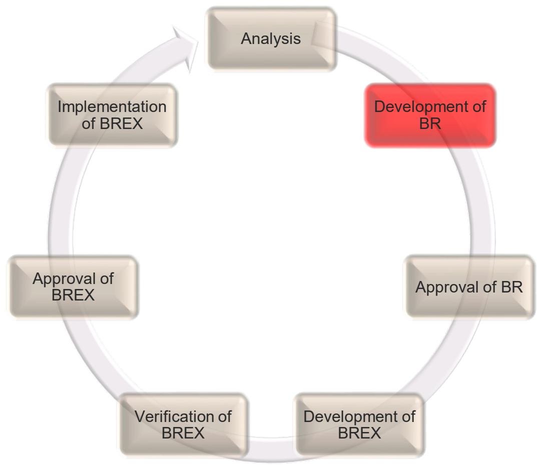


Chapter 3.9.3, Authoring – Warning, cautions and notes

- listItemPrefix (O), this element is used to indicate whether the list is a simple list or an unordered list, thus giving the prefix at the presentation. This attribute can have the following values (Refer to Chap 3.9.6.1):
  - "pf01" - no prefix (just an indenture)
  - "pf03" - prefix [-] used in notes only
  - "pf07" - prefix [●] used in warnings and cautions only

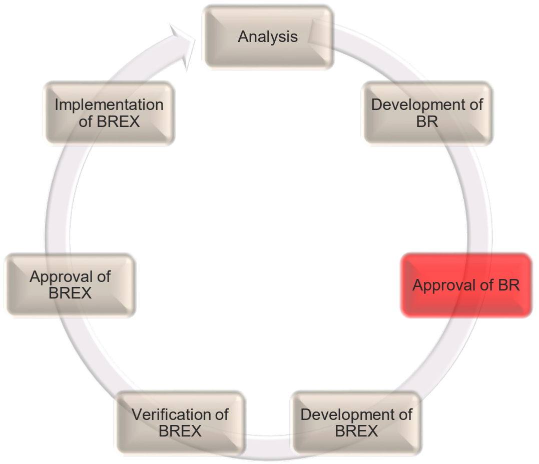
Note

The prefix value pf01, pf03 or pf07 has to be chosen instead of the default prefix of pf02.



BR approved, Chap 3.9.5.2.1.3

- **Simple or unordered lists**
  - The normal usage is to use the unordered list.
    - The **default value** of the attribute **listItemPrefix** for unordered list is “pf02”.
    - The simple list can only be used in exceptional cases. For the simple list the value “pf01” must be used.
  - The differences between simple and unordered lists are the layout:
    - simple list has no prefix
    - unordered list has the following sequence of prefix [-][•][-].

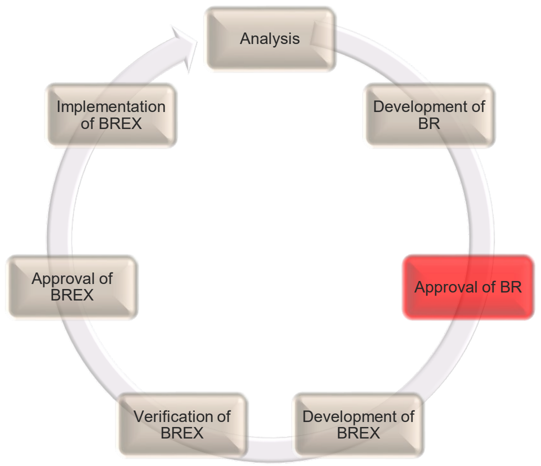


BR approved, Chap 3.9.6.1

- Table 29 `listItemPrefix` - Prefix for list items of random/unordered lists

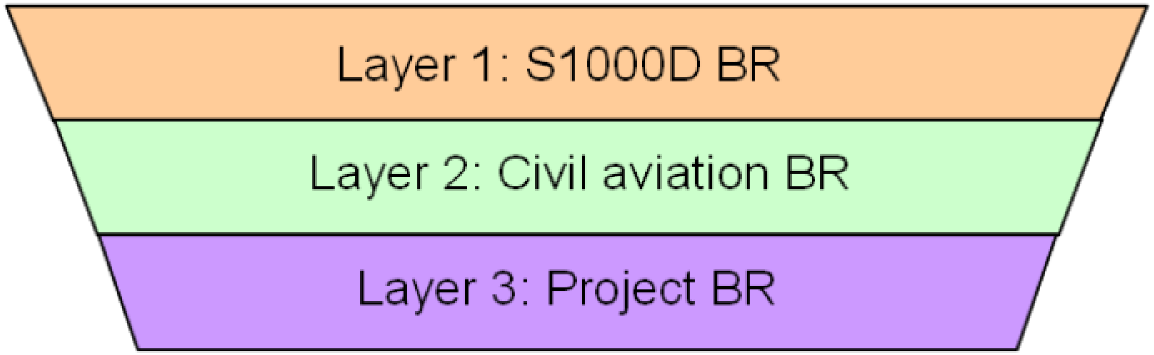
Allowable values	S1000D interpretation
pf01	Simple (No prefix, only indent)
pf02 (default value)	Unorder [-], [•], [-] (Depending on list level, prefix with short dash for first level, bullet for second, and short dash for third level - ISOpub: bull, dash)
pf03	prefix [-] used in notes only
pf07	prefix [•] used in warnings and cautions only

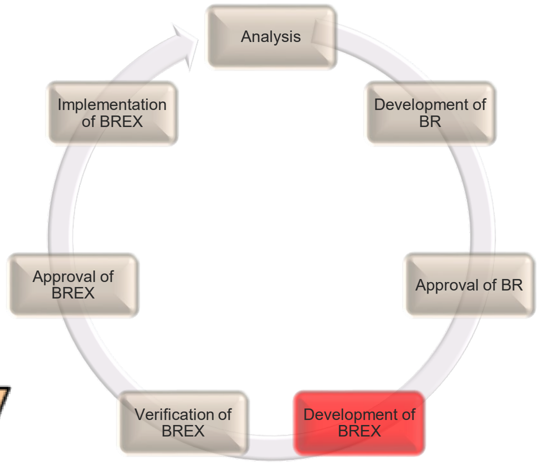
- No other values are allowed
  - *pf04 – pf06 or pf08 – pf99*



Layered BREX/Why use BREX data modules?

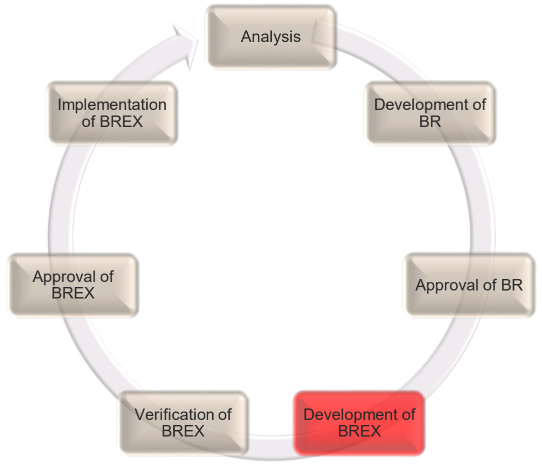
- Layers of BREX
  - Mandatory for a project BREX because S1000D default BREX must be used
  - More easy to control the development of the project BREX
- Why use BREX
  - Check that the project business rules are followed
  - More stringent than the S1000D XML schemas
  - Cost saving

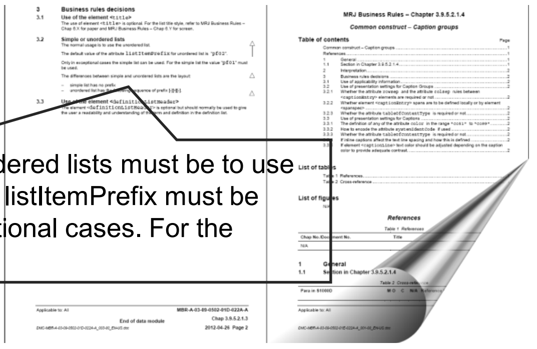




Development of BREX

- Can/should this BR be a BREX rule?
- Start to analyze Business Rules, ex:
  - N/A
  - Must not be used
  - The element <infoName> is mandatory
  - The attribute enterpriseCode must contain the permitted CAGE for the RPC
- Not all Business rules are possible to test
  - S2000M, PLCS and SCORM must not be used for project.

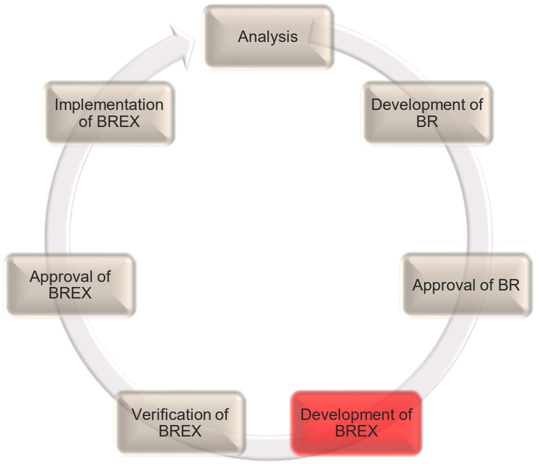




The default and normal usage for simple or unordered lists must be to use the unordered list. The default value for attribute listItemPrefix must be "pf02". The simple list can only be used in exceptional cases. For the simple list the value "pf01" must be used.

Creation of layered BREX data modules

- Required high skills in XML and XPath
- No special tools are required to create the BREX



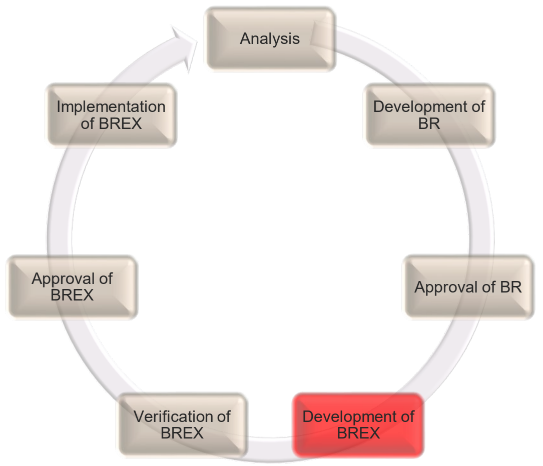
BREX DM – Creation

Business rule:

- The default and normal usage for simple or unordered lists must be to use the unordered list. The default value for attribute **listItemPrefix** must be "pf02". The simple list can only be used in exceptional cases. For the simple list the value "pf01" must be used.

BREX DM:

- <structureObjectRule>
  - <objectPath>//@listItemPrefix[**not**(ancestor::**note** or ancestor::**caution** or ancestor::**warning**)]</objectPath>
  - <objectUse> The default and normal …Text from BR …</objectUse>
  - <objectValue **valueForm**="single" **valueAllowed**="pf01">…</objectValue>
  - <objectValue **valueForm**="single" **valueAllowed**="pf02">...</objectValue>
- </structureObjectRule>



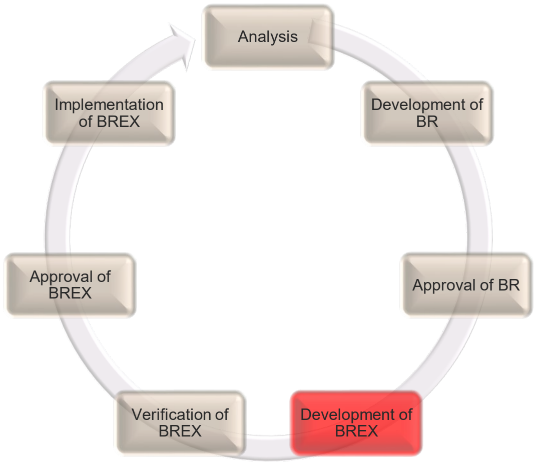
BREX DM – Creation, cont.

Chap 3.9.3:

- The prefix value pf01, pf03 (note) or pf07 (warning and caution) has to be chosen instead of the default prefix of pf02.

BREX DM:

- <structureObjectRule>
  - <objectPath>//@listItemPrefix[ancestor::**note**]</objectPath>
  - <objectUse> The default and normal …Text from BR …</objectUse>
  - <objectValue **valueForm**="single" **valueAllowed**="pf01">…</objectValue>
  - <objectValue **valueForm**="single" **valueAllowed**="pf03">...</objectValue>
- </structureObjectRule>



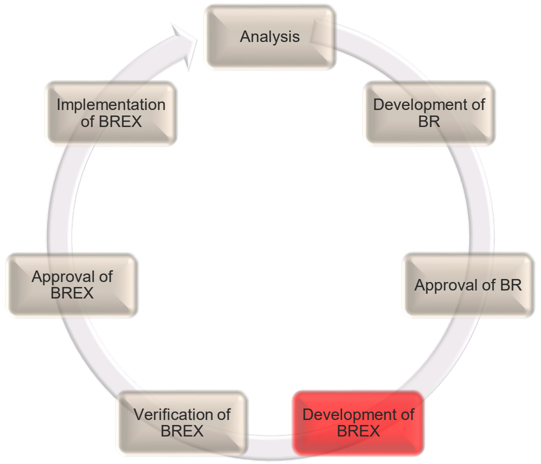
BREX DM – Creation, cont.

Chap 3.9.3:

- The prefix value pf01, pf03 (note) or pf07 (warning and caution) has to be chosen instead of the default prefix of pf02.

BREX DM:

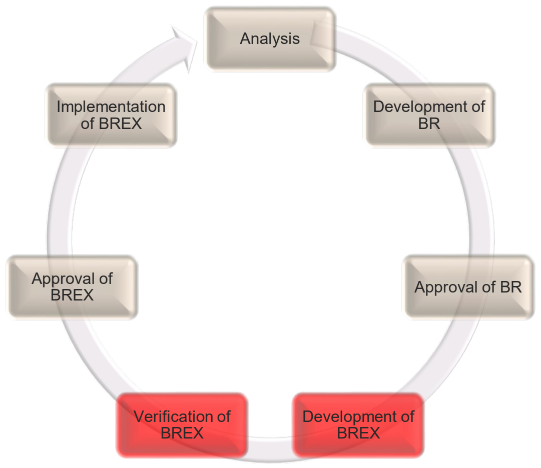
- <structureObjectRule>
  - <objectPath>//@listItemPrefix[ancestor::**caution** or ancestor::**warning**]</objectPath>
  - <objectUse>…</objectUse>
  - <objectValue **valueForm**="single" **valueAllowed**="pf01">…</objectValue>
  - <objectValue **valueForm**="single" **valueAllowed**="pf07">...</objectValue>
- </structureObjectRule>



Creation of test cases

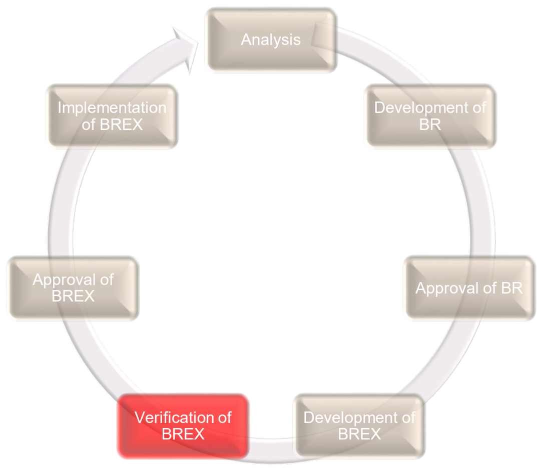
The test cases must include errors and no errors

- Outside the production system
  - For the development of the BREX
  - Import and export
  - Enables testing of all possible errors
- Inside the production system
  - Verification of the production system BREX functionality
  - Problem to create all possible errors
  - Import and export
  - Production of technical information



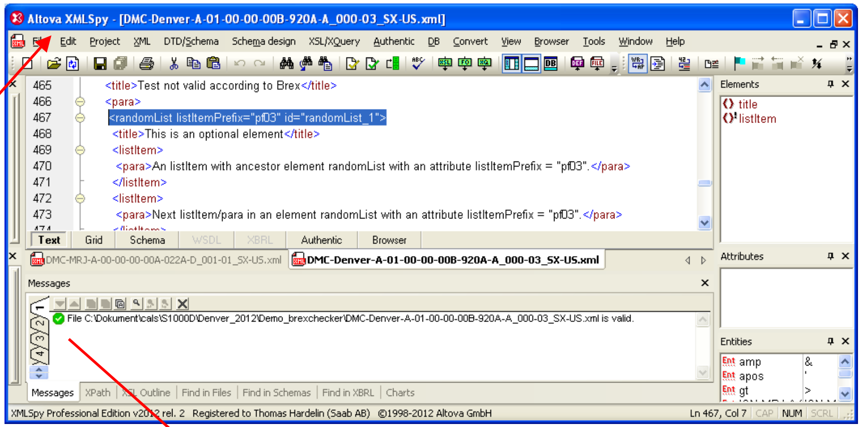
BREX checker

- Internal BREX checker
  - Inside the production system
- External BREX checker
  - Use for BREX development and test cases
  - Subcontractors
  - As a check before import/export
  - Check before delivery
  - Batch check



External BREX checker – What do You need?

- A BREX checker brex2xsl.xsl
- An XSLT Parser MSXSL.EXE
- S1000D Default BREX
- Project BREX DM
- DM to be tested


Altova XMLSpy - [DMC-Denver-A-01-00-00-00B-920A-A_000-03_SX-US.xml]

```
<title>Test not valid according to Brex</title>
<para>
 <randomList listItemPrefix="pf03" id="randomList_1">
  <title>This is an optional element</title>
  <listItem>
   <para>An listItem with ancestor element randomList with an attribute listItemPrefix = "pf03".</para>
  </listItem>
  <listItem>
   <para>Next listItem/para in an element randomList with an attribute listItemPrefix = "pf03".</para>
```

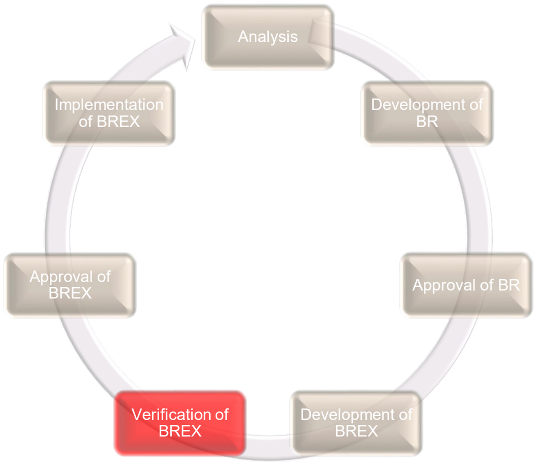
DMC-MRJ-A-00-00-00-00A-022A-D_001-01_SX-US.xml | DMC-Denver-A-01-00-00-00B-920A-A_000-03_SX-US.xml

Messages

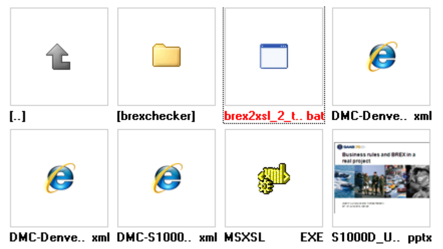
```
File C:\Dokument\cals\S1000D\Denver_2012\Demo_brexchecker\DMC-Denver-A-01-00-00-00B-920A-A_000-03_SX-US.xml is valid.
```

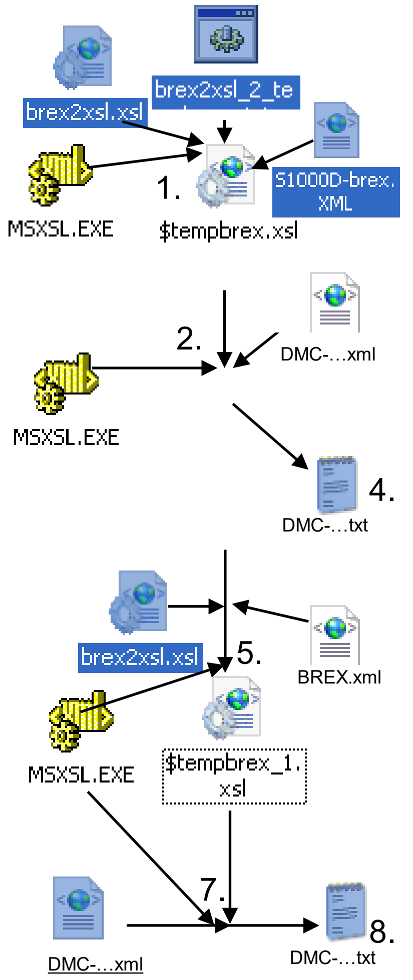
XMLSpy Professional Edition v2012 rel. 2 Registered to Thomas Hardelin (Saab AB) ©1998-2012 Altova GmbH   Ln 467, Col 7

Valid according to the corresponding S1000D XML Schema.



Demo of BREX with BREX checker





1. A bat file creates an new XSLT ($tempbrex.xsl) from S1000D BREX
2. Run test on all *.xml in current folder with the new XSLT ($tempbrex.xsl)
3. Pause on the bat file
4. Analyze errors (file size > 0 kb)
5. The bat file starts again and creates a new XSLT ($tempbrex_1.xsl) from project BREX
6. Pause on the bat file
7. Run a new test on all *.xml in current folder with the new XSLT ($tempbrex_1.xsl)
8. Analyze errors (file size > 0 kb)

Now it's time for a demo!

Lessons learned (SAAB experiences)

- Many parameters for a Business rules decision
- Need to have high skills in the specification to have knowledge about all affected chapters for a decision
- BREX doesn't parse against itself!
  - Only to the next layered level
- For some XML Schemas (issue 4.1) there are no <brexDmRef>
- Need to have good knowledge in the production system (limitations)
- Good communication between development of business rules and development of the BREX
- Re-decisions will be required when starting the development of the BREX
- A stand alone BREX checker for the development of BREX is required

Joakim.lundqvist@saabgroup.com

https://www.saab.com/