Action item 38-08:
Part 1 6.29 TEX, Other issues around loop control variables are missing.
See N0564 for JSF AV 198. & 199

Background

The requirements in 6.29 (for part 1) are:

- Do not modifying a loop control variable in the body of its associated loop body.
- Using a static analysis tool that identifies the modification of a loop control variable.
- Some languages, such as C and C++ do not explicitly specify which of the variables appearing in a loop header is the control variable for the loop. MISRA C [12] and MISRA C++ [16] have proposed algorithms for deducing which, if any, of these variables is the loop control …

In part 3, 6.29's guidance is:

- Apply the guidance of TR 24772-1 clause 6.29.5.
  Do not modify a loop control variable within a loop. Even though the capability exists in C, it is still considered to be a poor programming practice.

In the JSF C++ rules:

198. The initialization expression in a for loop will perform no actions other than to initialize the value of a single for loop parameter. Note that the initialization expression may invoke an accessor that returns an initial element in a sequence
199. The increment expression in a for loop will perform no action other than to change a single loop parameter to the next value for the loop

For MISRA C, the requirements for a for loop are:

- The loop counter shall not be of floating type
- The controlling expression shall not be invariant (applies to all loops)
- The for loop shall be well formed:
  - First expression
    - Shall be empty or
    - Assign a value to the loop counter or
    - Define and initialise the loop counter
  - Second expression
    - Shall be an expression with no persistent side effects and
    - Shall use the loop counter and (optionally) any loop control flags (*these terms are defined*) and
    - Shall not use any objects modified in the loop body, unless their essentially type is Boolean (*these are the loop control flags*)
  - Third expression
    - Shall be an expression whose only persistent side effect is to modify the loop counter and
    - Shall not use any object modified in the loop body

MISRA C++ adds:

- The third expression shall only modify the loop counter with one of: ++ -- += n or -= n where n is an integer value that is not modified in the body of the loop
- The second expression shall only use the loop counter in a relational expression, and if the loop counter is modified with += or -= this relationship shall not be equality or inequality (== or !=)

Proposal

It's not clear that the JSF rules apply to any other language than C and C++, as 'initialization expression' and 'increment expression' are essentially part of the C syntax. How would it apply to an Ada for loop for example?

```
for i in Integer range 1 .. 10 loop
```

As such, I propose that no modification to part 1 of the document is needed.

For C/C++, the JSF rules are a subset of the MISRA rules, and clearly the MISRA rules are more detailed than the guidelines currently in part 3. I suggest that expanding the guidance in part 3: 6.29 is considered as part of the review of the current draft.