ISO/IEC JTC 1
Information technology
Secretariat: ANSI (United States)

Document type:	Business Plan
Title:	Business Plan for JTC 1/SC 22, Programming languages, their environments and system software interfaces for the PERIOD COVERED: October 2015 – September 2016
Status:	This document is circulated for review and consideration the November 2016 JTC 1 meeting in Norway.
Date of document:	2016-10-03
Source:	SC 22 Chair
Expected action:	ACT
Action due date:	2016-11-07
Email of secretary:	lrajchel@ansi.org
Committee URL:	http://isotc.iso.org/livelink/livelink/open/jtc1

Business Plan for JTC 1/SC 22

Programming languages, their environments and system software interfaces

PERIOD COVERED: October 2015 – September 2016

Produced and Submitted by

Rex Jaeschke, Chair
(rex@RexJaeschke.com)

2016-09-20

1.0 Executive Summary

SC 22 continues to operate very well. While some WGs' membership and marketplace relevance are in decline, others are on the increase. All are operating in a manner appropriate to their needs.

WGs continue to monitor and to consider support for new[er] technologies, such as concurrency, multi-core processors, high-performance computing, and object-oriented and other methodologies.

SC 22 holds its Plenary on an annual basis, five to six weeks prior to the JTC 1 plenary. Between plenaries, formal business is carried out via letter ballots, and informal business is carried out via teleconferences and an email list. All committee documents are posted on Live Link. At most, one WG meets in conjunction with the plenary.

2.0 CHAIRMAN'S REMARKS

2.1 Market Requirements, Innovation

The classic programming languages, for which SC 22 is well known, remain popular with major development work going on in Fortran, C, C++, and Ada. Interest continues in the WG documenting vulnerabilities of various programming languages.

2.2 Accomplishments

This varies widely from one WG to another, from little activity besides DR processing in some, to high activity in others.

2.3 Resources

The C and C++ language WGs continue to have high participation, both in the number of members and NBs. The same is true for the WG documenting vulnerabilities of various programming languages. Participation in the other active WGs ranges from adequate to good, except for the COBOL WG, whose resources continue to diminish.

2.4 Competition and Cooperation

None of SC 22's projects has direct competition, and most WGs have active liaisons with related groups in their respective industries.

2.4 Issues to bring to JTC 1's Attention

See 5.0, below, regarding the limits of two TCs per edition, and no TCs after three years.

3.0 Working Groups

3.1 WG 4 — COBOL

Development and maintenance of ISO/IEC Standards and Technical Reports related to programming language COBOL.

COBOL remains one of the widely-used programming languages for both new development and enhancement of existing applications.

3.1.1 WG 4 Accomplishments

- Processing of defect reports.
- We are still waiting for implementations fully conforming to ISO/IEC 1989:2014.
- No project has been completed, underway, or cancelled over this period.

3.1.2 WG 4 Deliverables

None

3.1.3 WG 4 Risks, Opportunities and Issues

- Further decline in the resources is the main risk for the progress of WG 4 work. Most of the manpower for the technical work has retired.
- WG 4/OWG-2 had been established to investigate the future directions of COBOL standardization.

3.2 WG 5 — Fortran

The development and maintenance of ISO/IEC Fortran programming language standards.

Fortran is the language of choice for much scientific, engineering, and economic programming, particularly for very large programs that have evolved over many years.

3.2.1 WG 5 Accomplishments

- Processing of defect reports.
- Started work on a revision to the base language standard.

3.2.2 WG 5 Deliverables

- Type 2 TS on "Additional Parallel Features in Fortran"

3.2.3 WG 5 Risks, Opportunities and Issues

- A shortage of resources

3.3 WG 9 — Ada

The development and coordination of ISO standards and Technical Reports for Programming Language Ada.

Ada is the language of choice for important parts of the real-time, embedded systems community as well as aerospace and defense segments. Ada is also being used in railway and banking systems.

3.3.1 WG 9 Accomplishments

- Processing of defect reports.
- TC for ISO/IEC 8652
- Continued work on an update to the Ada Conformity Test Suite (ACATS).
- Began work on an update to the Ada Annex to WG23's TR set
- Began work on an update to the SPARK Part to WG 23 Technical Report on Vulnerabilities.
- Started looking into an update to TR 15942:2000 Guidance for the use of the Ada Programming Language in High Integrity Systems.

3.3.2 WG 9 Deliverables

- ISO/IEC 8652:2012/Cor.1: 2016 Programming Languages – Information Technology – Ada

3.3.3 WG 9 Risks, Opportunities and Issues

- Participation has been steady. Web conferencing has helped more people to participate.

3.4 WG 14 — C

Development and maintenance of ISO/IEC Standards related to the programming language C.

3.4.1 WG 14 Accomplishments

- Processing of defect reports.
- A new multipart Technical Specification is being considered:
  - Programming language C — Extensions for parallel programming — Part 1: Thread-based parallelism
  - Programming language C — Extensions for parallel programming — Part 2: Vector-based parallelism

3.4.2 WG 14 Deliverables

- TS on Floating-point extensions for C – Part 5:
- TS 17961:2013/COR 1, C Secure Coding Rules.
- TS 18661 part 3
- TS 18661 part 4

3.4.3 WG 14 Risks, Opportunities and Issues

- Participation has been good. Web conferencing has helped more people to participate.

3.5 WG 17 — Prolog

Development and maintenance of ISO/IEC standards related to programming language Prolog

Prolog is a niche language. It is extensively used by a small number of users mainly for applications in configuration, web and CGI generation, constraint handling and natural language. It is taught in a significant number of universities.

3.5.1 WG 17 Accomplishments

- A new TC
- PDTR on Definite Clause Grammars
- Amd to part II

3.5.2 WG 17 Deliverables

- ISO/IEC 13211-1. Programming language Prolog-part 1. General core ISO/IEC 13211-2. Programming language Prolog-part 2. Modules

3.5.3 WG 17 Risks, Opportunities and Issues

- Participation has been steady.

3.6 WG 21 — C++

Development and maintenance of ISO/IEC Standards, Technical Specifications, and Technical Reports related to the programming language C++.

ISO C++ remains a widely-used foundation technology, well-received in the marketplace.

Although C++ has long been a consistently popular language, since 2011 in particular it has enjoyed a renewed cycle of growth and investment in tools and platform support across the industry. This was driven primarily by the C++11 standard's completion at the same time as the industry saw a resurgence of interest in performance-efficient, hardware-efficient, and especially power-efficient systems programming capability for mobile devices, cloud data centers, high-performance financial systems, vector and GPGPU computing (via nonstandard extensions to C++ that we are now investigating standardizing), and other major growth sectors and environments.

3.6.1 WG 21 Accomplishments

Work is underway on the following:

- Revision of 14882
- PDTS draft of 19216: C++ Extensions for Networking
- Revision of 19568: C++ Extensions for Library Fundamentals
- Revision of 19570: C++ Extensions for Parallelism
- PDTS draft of 21425: C++ Extensions for Ranges
- PDTS draft of 21544: C++ Extensions for Modules

3.6.2 WG 21 Deliverables

- 19217: C++ Extensions for Concepts
- 19571: C++ Extensions for Concurrency
- 19841: C++ Extensions for Transactional Memory

3.6.3 WG 21 Risks, Opportunities and Issues

- WG21 has grown considerably over the past two years,

3.7 WG 23 — Programming Language Vulnerabilities

Development and maintenance of a TR series regarding “Guidance to Avoiding Vulnerabilities in

Programming Languages through Language Selection and Use"

The marketplace demands robust, secure software. Vulnerabilities are the antithesis of robust, secure software. Many of the attacks on software-based systems succeed because the computer language used did not prevent the attack vector, and did not warn the developer that the code being produced contained flaws that could be used to generate attacks.

3.7.1 WG 23 Accomplishments

- Working on slitting the TR into a core Part and separate language-specific Parts.

3.7.2 WG 23 Deliverables

None

3.7.3 WG 23 Risks, Opportunities and Issues

- Participation has been steady. Web conferencing has helped more people to participate.

4.0 JTC 1/SC 22 Dashboard 2016

Performance Indicators

	Systematic Reviews			Standards			
Year	Total Closed	Closed On time	% on time	Number Published	Avg time to Publish	# within timeframe	% within timeframe
2015	8	7	87.5%	1	32.40	1	100%
2016	1	1	100%				

5.0 Limitations on TC Number and Timespan

From SC 22's Plenary, September 2016:

Resolution 16-08: Request to Consider Modification of the JTC 1 Supplement on Technical Corrigenda

JTC 1/SC 22 notes that ISO policy limits the number of technical corrigenda to two, and that ISO/IEC Directives Part 1 specifies a limit of three years after which no technical corrigenda are allowed. These limitations pose general problems for standards maintenance. They pose additional problems for large standards (hundreds to thousands of pages) and those that have a long useful lifetime, such as those created and maintained by JTC 1/SC 22.

In the general case, a technical corrigendum is the only way to make a retroactive change to an existing edition of the standard. This allows business contracts pointing to a specific edition of a standard to receive the benefits of corrections and clarifications. As a result, the technical corrigendum is a necessary component for establishing and maintaining nondiscriminatory trade practices. Limiting the number and timing of technical corrigenda can have serious implications to users of JTC 1 standards.

Revising a 1,000-page standard is not a simple undertaking regarding time and resources, and is an error-prone process. Therefore, JTC 1/SC 22 prefers to not revise lengthy standards more frequently than technically necessary.

JTC 1/SC 22 agrees that technical corrigenda are only for corrections and resolutions of ambiguities, and not for adding new features. JTC 1/SC 22 believes that limiting the number and timing of technical corrigenda is neither a valid method nor an effective method for enforcing this policy, and imposes serious burdens on the maintenance of large standards.

Consequently, JTC 1/SC 22 urges JTC 1 to work with ISO to remove the limitation on timing and number of technical corrigenda.

The JTC 1/SC 22 Chair is requested to present this resolution to the November 2016 JTC 1 Plenary.

Unanimous Consent (All 9 NBs present: AT, CA, CN, DK, JP, KR, NL, UK, US)

The rationale for the above resolution follows, and will be presented at the JTC 1 Plenary:

1. Data Points re SC 22 and its Approach to Standards-Making
    a. SC 22's clients *demand* long-term stability; upgrading to new editions can be expensive and painful, especially for large code bases. As such, core standards have a 6–10-year lifecycle.
    b. Core standards are 600–1,000 pages long (one is 4,000 pages).
    c. Core standards are complex with many "moving parts" that are inter-related.
    d. Core standards often have demanding typesetting requirements, especially relating to mathematical notation. A single character in the wrong position or the wrong font can change the meaning of a standard.
    e. Rather than putting new features in a core standard revision, some WGs publish Technical Specifications, which allow the core standard to remain stable longer while valuable evidence is gained from implementation of trial-use features in TSs.
2. Limit of 3 Years for TCs on an edition
    a. It is possible that implementers will not have completed work on a product that conforms to a new standard within 3 years after a standard is adopted. So they might not be submitting many or any DRs during that time.

   b. It is very likely that users have not started work on conversion of existing code to a new implementation, or using that new version for new code, within 3 years after a standard is adopted. So they won't be submitting DRs.
   c. Given a 6–10-year standard lifecycle, that suggests allowing TCs up to 5–9 years.
3. Limit of 2 TCs per Edition
   a. Given our base standards' lifecycles, this means having no TCs in the second half or final 70% of many core standard's lives, which is not helpful.
   b. Requires batching DR resolutions into two large TCs, published less frequently, when it may well be more efficient and timely to have more and smaller ones, published more frequently, especially if the corrections are critical.
4. Forcing Revisions More Frequently
   a. Procurement contracts requiring conformance to a specific edition require conformance to all current and future TCs for that edition, whereas a new edition requires a new conformance contract. So forcing a new edition by revision bypasses those contracts.
   b. Reduces the ability to use TSs for trial implementation and use.