

Objective vs. Prescriptive Standards for Certification of Software-Intensive Systems

Jeff Joyce

Critical Systems Labs

SCC Meeting, Annapolis, 2 May 2011

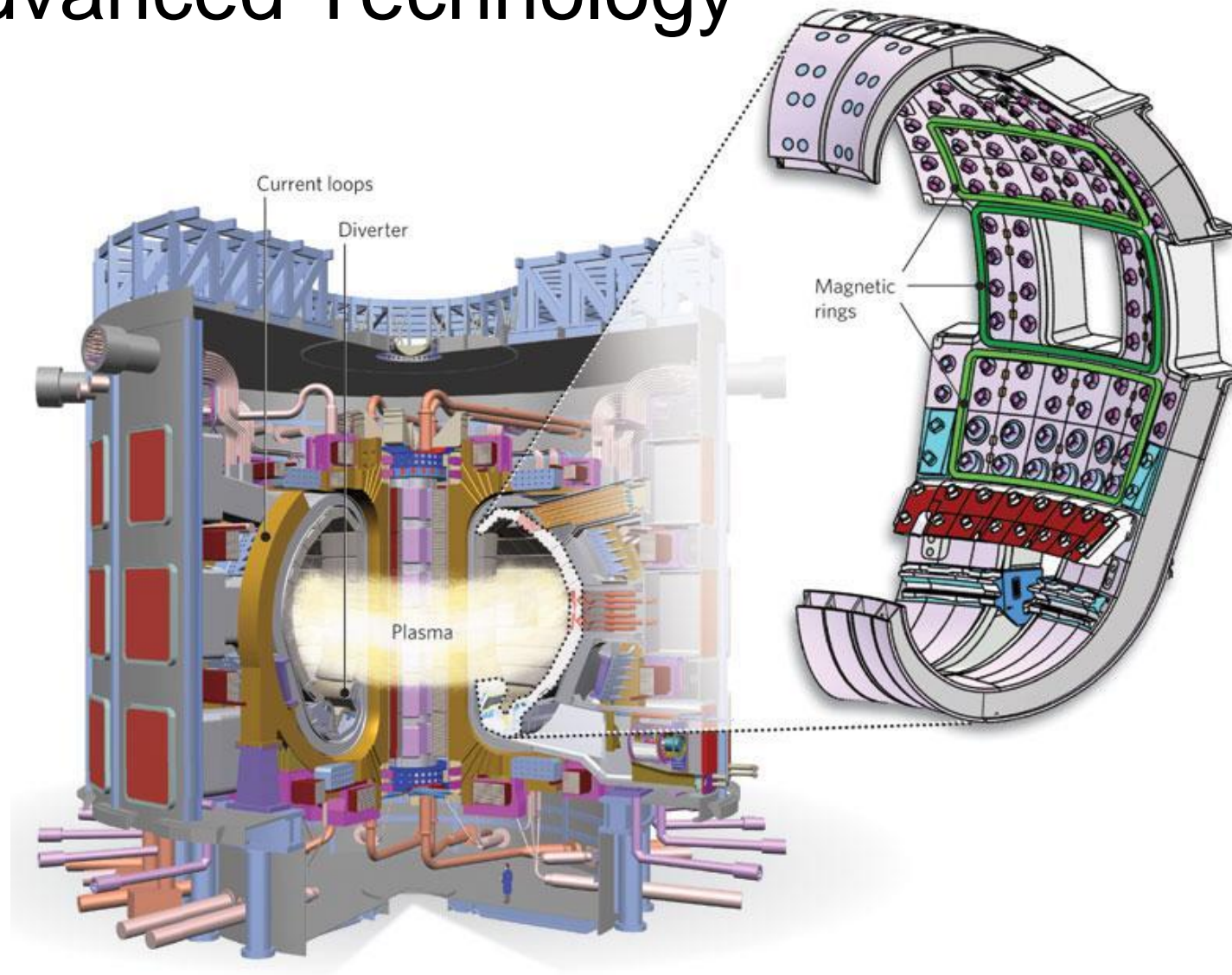
My Perspective

- Involved in a wide spectrum of industries that develop or depend on safety-related S/W including aerospace, defense, medical technology, rail signaling, automotive and high energy physics → **multi-industry**
- In addition to S/W, scope includes circuit design, firmware (e.g., VHDL), system engineering and even human factors → **multi-level**
- Beyond US and Canada, have long term working relationships with clients in Europe and Asia → **multi-cultural (global)**

My Perspective

- Used a variety of standards including MIL-STD 882, RTCA DO 178B, IEC 61508, CENELEC EN 50128 (rail signaling), ISO DIS 26262 (automotive), EUROCONTROL ESARR-6
- Familiar with many others, e.g., SAE ARP 4761, UK DEFSTAN 00-56, ISO 14971 (medical), IEC 61511 (process control)
- Participated in international working groups for both RTCA DO 178C (expected to replace DO 178B) and ISO DIS 26262 (automotive)


Advanced Technology



Evidence-focused Standards

- SCC work-plan calls for “evidence-focused” standards as a basis of certification
- Great, but what are you expecting to see in the standards about the nature of this evidence?
- To what extent are you comfortable with allowing compliance to depend on professional judgment and argument?

■ Is this what you mean by “evidence-focused”?

 The City of San Diego	<p align="center">MINIMUM REQUIREMENTS FOR Retaining Wall/Level Backfill</p> <p align="center">CITY OF SAN DIEGO DEVELOPMENT SERVICES 1222 FIRST AVENUE, MS 301 SAN DIEGO, CA 92101-4153 CALL (619) 446-5300 FOR APPOINTMENTS AND (619) 446-5000 FOR INFORMATION</p>	<p align="center">INFORMATION BULLETIN 221</p> <p align="center">AUGUST 2009</p>
<p>Construction of retaining walls, except those less than three feet high, measured from the top of the footing to the top of the wall and not supporting surcharge, requires a permit and is regulated by City of San Diego Municipal Code.</p> <p>Information Bulletin 221 outlines the city's requirements for retaining walls with level backfill. Information Bulletin 222 describes retaining walls with sloping backfill. These bulletins are intended to provide a simple alternative to designing minor retaining walls, but should be used only where appropriate soil condition at the site. See Section VII. SOIL.</p> <p>For information on how to obtain a permit for a retaining wall, see Information Bulletin 220.</p>		
<p>I. ZONING REGULATIONS Retaining walls heights are also regulated by</p>	<p>14, Article 2, the height of setbacks and</p>	<p>Documents Referenced in this Information Bulletin</p> <ul style="list-style-type: none"> • 2007 California Building Code, (CBC) • San Diego Municipal Code, (SDMC) • Information Bulletin 220, How to Obtain a Permit for a Retaining Wall/Fence • Information Bulletin 222, Minimum Requirements for Retaining Wall/Sloping Backfill
<p>IV. MASONRY BLOCKS Concrete masonry units shall be of sizes shown on drawings and conform to ASTM C90 (CBC 2103.1) Medium Weight Units with maximum linear shrinkage of 0.06%, F'm=1,500 psi grouted solid reinforced cells.</p> <p>All head and bed joints shall be 3/8" thick. Bed joints of the starting course over the concrete foundation may be between 1/4" and 3/4". (ACI 530.1-05 section 3.3B)</p>	<p>14, Article 2, the height of setbacks and</p> <p>was shall not 142.0340(b)).</p> <p>imum height the required two retaining in horizontal is upper wall.</p> <p>num height of required side ing walls are nial distance wall. (SDMC</p> <p>n 5 feet may</p>	<p>III. CAL/OSHA PERMIT/WAIVER A CAL/OSHA construction activity permit is required for construction of trenches or excavations which are five feet or deeper and into which a person is required to descend. For more information please contact:</p> <p>Cal/OSHA Enforcement Unit district office 7575 Metropolitan Drive, Ste. 207 San Diego 92108 (619) 767-2280 Fax (619) 767-2299</p>
<p>For the purpose of designing the wall in this information bulletin, wall height is measured from the top of the footing to the top of the wall. Walls not shown in Tables A on page 3 must be designed specifically for the existing conditions. The walls shown here are designed to retain only level backfill. No building foundation, retaining wall, driveway, parking, fence, or other potential source of loading on the upper level is allowed within a distance equal to the height of the wall. See figure 1.</p>	<p>num height of required side ing walls are nial distance wall. (SDMC</p> <p>n 5 feet may</p>	<p>IV. MASONRY BLOCKS Concrete masonry units shall be of sizes shown on drawings and conform to ASTM C90 (CBC 2103.1) Medium Weight Units with maximum linear shrinkage of 0.06%, F'm=1,500 psi grouted solid reinforced cells.</p> <p>All head and bed joints shall be 3/8" thick. Bed joints of the starting course over the concrete foundation may be between 1/4" and 3/4". (ACI 530.1-05 section 3.3B)</p> <p>No special inspection is required for retaining walls up to 6 feet in height.</p>
<p>Printed on recycled paper. Visit our web site at www.sandiego.gov/development_services. Upon request, this information is available in alternative formats for persons with disabilities.</p>	<p>num height of required side ing walls are nial distance wall. (SDMC</p> <p>n 5 feet may</p>	<p>V. SPECIFICATIONS</p> <p>A. CONCRETE Concrete for footings must have a minimum compressive strength of 2,500 psi at 28 days. (CBC 1805.4.2.1). Cement shall conform to ASTM-C150 (ACI 318-05 section 3.2).</p> <p>Note: Plastic (Stucco) cement ASTM C 1328 is not permitted in retaining walls located in Seismic Design Category D.</p> <p>B. MORTAR The mortar mix must have a compressive strength equal to 1,800 psi minimum (CBC Table 2105.2.2.1.2). Mortar for use in masonry</p>

■ Or do you mean something less prescriptive?

IEC 61508 Part 3

Table B.1 – Design and coding standards
(referenced by table A.4)

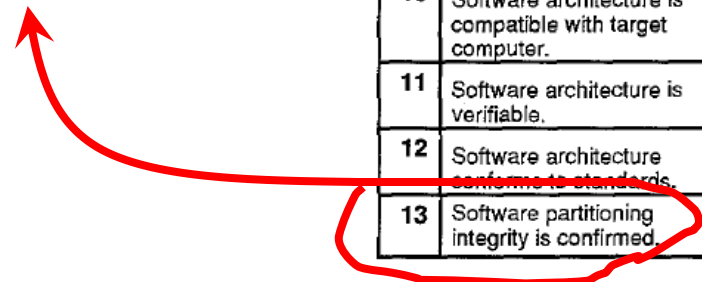
	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Use of coding standard	C.2.6.2	HR	HR	HR	HR
2	<u>No dynamic objects</u>	C.2.6.3	R	HR	HR	HR
3a	<u>No dynamic variables</u>	C.2.6.3	---	R	HR	HR
3b	Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4	<u>Limited use of interrupts</u>	C.2.6.5	R	R	HR	HR
5	<u>Limited use of pointers</u>	C.2.6.6	---	R	HR	HR
6	<u>Limited use of recursion</u>	C.2.6.7	---	R	HR	HR
7	No unconditional jumps in programs in higher level languages	C.2.6.2	R	HR	HR	HR
<p>NOTE – Measures 2 and 3a do not need to be applied if a compiler is used which ensures that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which inserts runtime checks for the correct online allocation of memory.</p>						
<p>* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.</p>						

Table A-4
Verification Of Outputs of Software Design Process

	Objective		Applicability by SW Level				Output		Control Category by SW level				
	Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D	
1	Low-level requirements comply with high-level requirements.	6.3.2a	●	●	○		Software Verification Results	11.14	②	②	②		
2	Low-level requirements are accurate and consistent.	6.3.2b	●	●	○		Software Verification Results	11.14	②	②	②		
3	Low-level requirements are compatible with target computer.	6.3.2c	○	○			Software Verification Results	11.14	②	②			
4	Low-level requirements are verifiable.	6.3.2d	○	○			Software Verification Results	11.14	②	②			
5	Low-level requirements conform to standards.	6.3.2e	○	○	○		Software Verification Results	11.14	②	②	②		
6	Low-level requirements are traceable to high-level requirements.	6.3.2f	○	○	○		Software Verification Results	11.14	②	②	②		
7	Algorithms are accurate.	6.3.2g	●	●	○		Software Verification Results	11.14	②	②	②		
8	Software architecture is compatible with high-level requirements.	6.3.3a	●	○	○		Software Verification Results	11.14	②	②	②		
9	Software architecture is consistent.	6.3.3b	●	○	○		Software Verification Results	11.14	②	②	②		
10	Software architecture is compatible with target computer.	6.3.3c	○	○			Software Verification Results	11.14	②	②			
11	Software architecture is verifiable.	6.3.3d	○	○			Software Verification Results	11.14	②	②			
12	Software architecture conforms to standards.	6.3.3e	○	○	○		Software Verification Results	11.14	②	②	②		
13	Software partitioning integrity is confirmed.	6.3.3f	●	○	○	○	Software Verification Results	11.14	②	②	②	②	

DO-178B:
Table A-4

Objective A-4.13
Software partitioning is confirmed



- Both examples refer to properties of the product (rather than the process) and are motivated the same underlying concern

Table A-4
Verification Of Outputs of Software Design Process

	Objective		Applicability by SW Level				Output		Control Category by SW level			
	Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D
1	Low-level requirements comply with high-level requirements.	6.3.2a	●	●	○	○	Software Verification Results	11.14	②	②	②	②
2	Low-level requirements are accurate and consistent.	6.3.2b	●	●	○	○	Software Verification Results	11.14	②	②	②	②
3	Low-level requirements are compatible with target computer.	6.3.2c	○	○	○	○	Software Verification Results	11.14	②	②	②	②
4	Low-level requirements are verifiable.	6.3.2d	○	○	○	○	Software Verification Results	11.14	②	②	②	②
5	Low-level requirements conform to standards.	6.3.2e	○	○	○	○	Software Verification Results	11.14	②	②	②	②
6	Low-level requirements are traceable to high-level requirements.	6.3.2f	○	○	○	○	Software Verification Results	11.14	②	②	②	②
7	Algorithms are accurate.	6.3.2g	●	●	○	○	Software Verification Results	11.14	②	②	②	②
8	Software architecture is compatible with high-level requirements.	6.3.3a	●	○	○	○	Software Verification Results	11.14	②	②	②	②
9	Software architecture is consistent.	6.3.3b	●	○	○	○	Software Verification Results	11.14	②	②	②	②
10	Software architecture is compatible with target computer.	6.3.3c	○	○	○	○	Software Verification Results	11.14	②	②	②	②
11	Software architecture is verifiable.	6.3.3d	○	○	○	○	Software Verification Results	11.14	②	②	②	②
12	Software architecture conforms to standards.	6.3.3e	○	○	○	○	Software Verification Results	11.14	②	②	②	②
13	Software partitioning integrity is confirmed.	6.3.3f	●	○	○	○	Software Verification Results	11.14	②	②	②	②

“Software partitioning is confirmed”

Table B.1 – Design and coding standards (referenced by table A.4)

Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1 Use of coding standard	C.2.6.2	HR	HR	HR	HR
2 No dynamic objects	C.2.6.3	R	HR	HR	HR
3a No dynamic variables	C.2.6.3	---	R	HR	HR
3b Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4 Limited use of interrupts	C.2.6.5	R	R	HR	HR
5 Limited use of pointers	C.2.6.6	---	R	HR	HR
6 Limited use of recursion	C.2.6.7	---	R	HR	HR
7 No unconditional jumps in programs in higher level languages	C.2.6.2	R	HR	HR	HR

NOTE – Measures 2 and 3a do not need to be applied if a compiler is used which ensures that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which inserts runtime checks for the correct online allocation of memory.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

“No dynamic objects, no dynamic variables, limited use of interrupts, limited use of pointers, limited use of recursion”

- However, one is expressed in terms of an objective while the other is prescriptive

A Fundamental Question for the SCC

- Should an evidence-focused S/W standard be objective like this or prescriptive like this ?

Table A-4
Verification Of Outputs of Software Design Process

Objective	Applicability by SW Level				Output	Control Category by SW level							
	Description	Ref.	A	B		C	D	Description	Ref.	A	B	C	D
1	Low-level requirements comply with high-level requirements.	6.3.2a	●	●	○	○	Software Verification Results	11.14	②	②	②	②	
2	Low-level requirements are accurate and consistent.	6.3.2b	●	●	○	○	Software Verification Results	11.14	②	②	②	②	
3	Low-level requirements are compatible with target computer.	6.3.2c	○	○	○	○	Software Verification Results	11.14	②	②			
4	Low-level requirements are verifiable.	6.3.2d	○	○	○	○	Software Verification Results	11.14	②	②			
5	Low-level requirements conform to standards.	6.3.2e	○	○	○	○	Software Verification Results	11.14	②	②	②	②	
6	Low-level requirements are traceable to high-level requirements.	6.3.2f	○	○	○	○	Software Verification Results	11.14	②	②	②	②	
7	Algorithms are accurate.	6.3.2g	●	●	○	○	Software Verification Results	11.14	②	②	②	②	
8	Software architecture is compatible with high-level requirements.	6.3.3a	●	○	○	○	Software Verification Results	11.14	②	②	②	②	
9	Software architecture is consistent.	6.3.2b	●	○	○	○	Software Verification Results	11.14	②	②	②	②	
10	Software architecture is compatible with target computer.	6.3.3c	○	○	○	○	Software Verification Results	11.14	②	②			
11	Software architecture is verifiable.	6.3.3d	○	○	○	○	Software Verification Results	11.14	②	②			
12	Software architecture conforms to standards.	6.3.3e	○	○	○	○	Software Verification Results	11.14	②	②	②	②	
13	Software partitioning integrity is confirmed.	6.3.3f	●	○	○	○	Software Verification Results	11.14	②	②	②	②	②

OR

Table B.1 – Design and coding standards (referenced by table A.4)

Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1 Use of coding standard	C.2.6.2	HR	HR	HR	HR
2 No dynamic objects	C.2.6.3	R	HR	HR	HR
3a No dynamic variables	C.2.6.3	---	R	HR	HR
3b Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4 Limited use of interrupts	C.2.6.5	R	R	HR	HR
5 Limited use of pointers	C.2.6.6	---	R	HR	HR
6 Limited use of recursion	C.2.6.7	---	R	HR	HR
7 No unconditional jumps in programs in higher level languages	C.2.6.2	R	HR	HR	HR

NOTE – Measures 2 and 3a do not need to be applied if a compiler is used which ensures that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which inserts runtime checks for the correct online allocation of memory.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

“Software partitioning is confirmed”

“No dynamic objects, no dynamic variables, limited use of interrupts, limited use of pointers, limited use of recursion”

Objective Approach: arguments for

- Allows experienced and knowledgeable experts to decide on the most effective way to achieve desired properties such as safety, reliability, availability, correctness
- Accommodates new techniques and methods
- Holistic –compatible with the increasing recognition of the fact that problems with complex software systems are not merely failures of individual components, e.g., feature interaction problems
- Keeps the effort focus on the overall goal (e.g. safety), rather than ticking off boxes

Objective Approach: against

- Too open-ended, vulnerable to ignorance or abuse, more susceptible to confirmation bias
- Harder to plan accurately, especially if approach allows tactical decisions to adjust priorities and resources allocation as understanding of the system and its sources of risk deepens
- May not be entirely compatible with some legal systems in regard to product liability, in particular, legal systems that rely more on codification than case law

Prescriptive Approach: arguments for

- More like traditional approach to engineering certification/regulation, e.g., building codes
- Less vulnerable to ignorance or abuse, confirmation bias
- Easier to plan, e.g., just need to do X, Y and Z
- More compatible with some legal systems

Prescriptive Approach: against

- Tick-box mentality
- Inhibits use of new (and possibly better) techniques



: “We plan to use formal proof to show that the kernel provides temporal and spatial partitioning”

“Sorry mate, but IEC 61508 says ...

Table B.1 – Design and coding standards (referenced by table A.4)

Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1 Use of coding standard	C.2.6.2	HR	HR	HR	HR
2 No dynamic objects	C.2.6.3	R	HR	HR	HR
3a No dynamic variables	C.2.6.3	---	R	HR	HR
3b Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4 Limited use of interrupts	C.2.6.5	R	R	HR	HR
5 Limited use of pointers	C.2.6.6	---	R	HR	HR
6 Limited use of recursion	C.2.6.7	---	R	HR	HR
7 No unconditional jumps in programs in higher level languages	C.2.6.2	R	HR	HR	HR

NOTE – Measures 2 and 3a do not need to be applied if a compiler is used which ensures that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which inserts runtime checks for the correct online allocation of memory.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

”

Prescriptive Approach: against (continued)

- Excludes perfectly valid designs, e.g., “no recursion”
- Just doing X, Y and Z might be enough to stop a retaining wall from collapsing, but hard to imagine ever making a comprehensive list to assure anything about S/W (unless it’s a list of objectives)
- Not compatible with some legal systems in regard to product liability
- Sometime too much of a laundry list of favorite techniques of individual members in the group that developed the standard, influenced considerably by politics and status

Prescriptive Approach: against (continued)

- Often involves making a list of “safety protection functions” at a very early stage in development and then focuses resources on ensuring the reliability of these safety protection functions in proportion to the assessed level of risk
 - This *might* work in the case of “mature technology” where the hazards are well known at the start
 - But this does not work well for “young technology”

Supply Chain Outsourcing

- OEM contracts first tier supplier X who in turn sub-contracts part of the task to Y who collaborates with Z ...
- As you follow the supply chain, there is a decreasingly likelihood that everyone contributing has sufficient knowledge and experience to properly use an objective oriented standard such as DO 178B
- This is a deep concern within some industries that should be considered by the SCC

Depends what is meant by “objectives”

- A prescriptive standard might refer to objectives, but does this mean that it’s objective after all?
- For example, in IEC 61508-3 ...

7.2.1 Objectives

7.2.1.1 The first objective of the requirements of this subclause is to specify the requirements for software safety in terms of the requirements for software safety functions and the requirements for software safety integrity.

7.2.1.2 The second objective of the requirements of this subclause is to specify the requirements for the software safety functions for each E/E/PE safety-related system necessary to implement the required safety functions.

7.2.1.3 The third objective of the requirements of this subclause is to specify the requirements for software safety integrity for each E/E/PE safety-related system necessary to achieve the safety integrity level specified for each safety function allocated to that E/E/PE safety-related system.

Prescriptive → Process-Focused

- In case of building codes for retaining walls, we can have a standard that is both evidence-focused and depends relatively

IV. MASONRY BLOCKS

Concrete masonry units shall be of sizes shown on drawings and conform to ASTM C90 (CBC 2103.1) Medium Weight Units with maximum linear shrinkage of 0.06%, F'm=1,500 psi grouted solid reinforced cells.

All head and bed joints shall be 3/8" thick. Bed joints of the starting course over the concrete foundation may be between 1/4" and 3/4". (ACI 530.1-05 section 3.3B)

little on *professional judgment* and argument

- But S/W systems are not like retaining walls and, in general, it seems impossible to have a useful certification standard for S/W that is evidence-focused without a need for professional judgment and argument

CENELEC EN 50128

Table A.2 – Software Requirements Specification (clause 8)

TECHNIQUE/MEASURE	Ref	SWS ILO	SWS IL1	SWS IL2	SWS IL3	SWS IL4
1. Formal Methods including for example CCS, CSP, HOL, LOTOS, OBJ, Temporal Logic, VDM, Z and B	B.30	-	R	R	HR	HR
2. Semi-Formal Methods	D.7	R	R	R	HR	HR
3. Structured. Methodology including for example JSD, MASCOT, SADT, SDL, SSADM, and Yourdon.	B.60	R	HR	HR	HR	HR
Requirements <ol style="list-style-type: none"> The Software Requirements Specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application. The table reflects additional requirements for defining the specification clearly and precisely. One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used. 						

R means “recommended”

HR means “highly recommended”

For S/W, a choice between ...

OR

- Evidence-focused
- Objective-oriented
- Depends on professional judgment and argument
 - ... comes with worries about abuse, ignorance, confirmation bias, ...

- Process-focused
 - but amounts to “Circumstantial” evidence only
- Prescriptive
- Relatively little need for professional judgment and argument

Paddling Upstream

- It is hard work to formulate truly meaningful objectives that serve as a basis for certification
- It is hard to do this as an individual and even harder to this in a group, especially an international working group with a mixture of non-technical factors, i.e.,
 - individual interests and priorities
 - business interests and priorities
 - national interests and priorities

What works well for CSL

- When helping clients developing internal organizational standards and guidance, we have found the following format to be effective
 1. Objective (normative)
 - A clear statement of the objective that refers to desired quality or quantity
 2. Assessment Criteria (recommended)
 - A list of criteria that should be used to determine the extent to which the objective has been satisfied
 3. Methods and Techniques (informative only)
 - What would be typically found in a prescriptive standard

Conclusions and Recommendations

- SCC ought to be clear about the extent to which evidence-focused standard should be objective rather than prescriptive
- Objective standards entail the need for professional judgment and argument
 - ... which comes with such worries as ignorance, abuse and confirmation bias
- However, the alternative (prescriptive approaches) has overwhelming disadvantages that make them unsuitable for anything except possibly very simple S/W in the context of mature technology