

The Absence of Accessibility in Software Sizing Frameworks: A Gap Analysis of UCP, COSMIC, and IFPUG SNAP*

Katarzyna MACIEJEWSKA

Polish-Japanese Academy of Information Technology (PJATK),
Department of Software Engineering, Warsaw, Poland

Correspondence should be addressed to: Katarzyna MACIEJEWSKA, kata@pjwstk.edu.pl

*Presented at the 47th IBIMA International Conference, 29-30 June 2026, Madrid, Spain

Abstract

Accurate software cost estimation depends on sizing methods that capture the full scope of project requirements, including non-functional requirements such as accessibility. This study examines whether three established sizing frameworks, Use Case Points (UCP), COSMIC Function Points, and IFPUG SNAP, operationalise accessibility as a non-functional requirement. Accessibility, commonly operationalised through conformance with WCAG guidelines (W3C, 2023), generates specific, verifiable implementation effort across multiple development phases, yet it is rarely treated as a formal cost driver in sizing models. The study uses document analysis of the normative sources for each framework. The findings are consistent across all three: none provides an explicit classification mechanism, counting rules, complexity parameters, or a calculation procedure for accessibility requirements. In SNAP, the most relevant framework, as it is specifically designed for non-functional requirements, accessibility appears in the ISO 25010 quality model as a sub-characteristic of Usability and in the normative document as a one-time disclaimer, but is not operationalised as a distinct, reproducible measurement construct. The issue is not that accessibility cannot be associated with existing categories, but that the Assessment Practices Manual provides no explicit, reproducible procedure for doing so. Two of the three frameworks have formal structural extension points that could accommodate accessibility: COSMIC through its open NFR slots, and SNAP through its ISO 25010 mapping structure. UCP has no formally documented extension mechanism, though its factor list is not explicitly closed to additions. The absence of accessibility from these frameworks may contribute to systematic underestimation of accessibility costs in software projects.

Keywords: software cost estimation, web accessibility, WCAG, Use Case Points

Introduction

Software sizing methods are used to estimate the effort and cost of software projects by measuring software size based on functional and non-functional requirements. This study examines whether software sizing methods represent accessibility as a non-functional requirement. Accessibility, commonly operationalised through conformance with WCAG guidelines, is a growing concern in software development. In the European Union, it is regulated by Directive 2016/2102/EU (Web Accessibility Directive) and the European Accessibility Act (Directive 2019/882/EU, applying to new products and services from June 2025), with EN 301 549 v3.2.1 as the harmonised technical standard referenced in public procurement. If established sizing methods do not treat it as a cost driver, then projects with accessibility requirements may be systematically underestimated.

Three sizing approaches are considered in this study: Use Case Points (UCP), COSMIC Function Points, and IFPUG SNAP. Software sizing and functional size measurement are recognised as a critical foundation for

investment and project decisions in software development (Czarnacka-Chrobot, 2013a; 2013b). Accessibility warrants treatment as a distinct non-functional requirement for three reasons. First, WCAG success criteria are specific, testable, and externally defined: they define precise, binary or graduated pass/fail conditions that can be systematically verified through automated tools or manual inspection, making them well-suited to systematic measurement. Second, achieving WCAG conformance requires dedicated effort across multiple development phases, requirements analysis, design decisions, implementation, and testing, and this effort is not captured by functional sizing alone. Third, the presence of accessibility requirements in a project is not incidental: it is increasingly determined by legal obligations, procurement specifications, or organisational standards. These characteristics justify treating accessibility as a distinct, measurable requirement category rather than an implicit quality concern.

Throughout this article, "operationalised" means that a sizing framework provides an explicit classification mechanism, counting rules, complexity parameters, and a calculation procedure sufficient to produce a numeric size estimate for requirements of that type. The presence of a term in documentation, without associated counting rules, does not constitute operationalisation in this sense.

UCP produces a software size estimate by multiplying unadjusted use case points by two correction factors: the Technical Complexity Factor (TCF) and the Environmental Complexity Factor (ECF). COSMIC addresses non-functional requirements through a dedicated practitioner guide that complements its functional measurement standard. SNAP is a purpose-built method for sizing non-functional requirements, now recognised as an international standard (ISO/IEC/IEEE 32430:2025). Together, these three frameworks represent structurally distinct and established approaches to software sizing.

The study uses document analysis of the normative sources for each framework: Karner (1993) for UCP (with Ochodek, Nawrocki and Kwarciak (2011) providing supplementary empirical context on factor simplification), the COSMIC NFR Practitioner's Guide v1.0 (Abran et al., 2020) for COSMIC, and the SNAP Assessment Practices Manual v2.4 (IFPUG) as confirmed by Buglione and Di Cola (2025) and ISO/IEC/IEEE 32430:2025 for SNAP. The analysis is limited to these normative documents and does not examine how the frameworks are applied in practice.

For each framework, the analysis asks three questions. First, is accessibility or WCAG-based conformance explicitly named as a non-functional requirement category? Second, does the framework have a structural extension point where a new category could be added without fundamental redesign? Third, does accessibility appear in any illustrative examples or counting guidance, even if not as a formal category? The absence of accessibility across all three dimensions is what this study calls the gap.

Overview of Software Sizing Methods

This section examines three established software sizing frameworks: Use Case Points (UCP), COSMIC Function Points, and IFPUG SNAP, and evaluates whether accessibility appears as an explicit category, in illustrative examples, or in counting guidance.

Use Case Points (UCP)

Use Case Points were introduced by Karner (1993) as a method for estimating software size from UML use case models. The method produces a size estimate through two multiplicative correction factors applied to a base count of use case complexity: the Technical Complexity Factor (TCF) and the Environmental Complexity Factor (ECF). TCF is the factor relevant to this analysis, as it encodes assumptions about the technical characteristics of the software being sized. It is calculated as:

$TCF = 0.6 + (0.01 \times \text{sum of 13 weighted Technical Factors, } T1-T13), \text{ giving } UCP = UUCP \times TCF \times ECF.$

Table 1 lists the 13 Technical Factors with their weights as defined by Karner (1993).

Table 1 UCP Technical Complexity Factors and weights (Karner, 1993), mapped against accessibility relevance

Technical Factor	Weight (Karner, 1993)	Addresses accessibility?
T1 Distributed System	2	No
T2 Response/Performance	1	No
T3 End-user Efficiency	1	Partially (general usability only)
T4 Complex Processing	1	No
T5 Reusable Code	1	No
T6 Easy to Install	0.5	No
T7 Easy to Use	0.5	Partially (general usability only)
T8 Portable	2	No
T9 Easy to Change	1	No
T10 Concurrent	1	No
T11 Security Features	1	No
T12 Access for Third Parties	1	No
T13 Special Training Required	1	No

Note: Technical Factor names and weights reproduced from Karner (1993); the "Addresses accessibility?" column reflects the present study's own analysis.

None of T1–T13 explicitly addresses accessibility or WCAG conformance. T7 ("Easy to Use") is the closest conceptually, but it concerns general usability, not accessibility for users with disabilities. Two additional factors may appear relevant at first glance. T3 ("End-user Efficiency") addresses performance and interaction speed for a general user, not the specific requirements of users with disabilities as defined by WCAG. T11 ("Security Features") concerns data protection and access control, an entirely different quality dimension. Neither factor provides a basis for sizing accessibility requirements. A later study by Ochodek, Nawrocki and Kwarciak (2011) examined whether all 13 Technical Factors are statistically significant for effort prediction and found that some could be removed without loss of accuracy (Information and Software Technology, 53(3), 200–213). However, this addresses a different question. Their study examined the statistical significance of the existing factors, whereas the present study examines whether accessibility is represented in the factor space at all. A factor can be missing from the TCF list regardless of whether the remaining factors are later found to be unnecessary or could be removed.

COSMIC Function Points

COSMIC is an ISO standard (ISO 19761) based on four types of data movements: Entry, Exit, Read, and Write. It is focused on functional user requirements by design. Research has shown that COSMIC-based effort models can incorporate technological cost factors (Czarnacka-Chrobot, 2013b), suggesting that estimation approaches built around COSMIC are not necessarily limited to functional size alone. The question of non-functional requirements is addressed more directly in the COSMIC "Non-Functional Requirements and COSMIC Sizing: Practitioner's Guide" (Abran et al., 2020). The guide identifies 14 standard system-level NFR categories drawn from ISO, IEEE, and ECSS standards: Reliability, Maintainability, Interfaces, Portability, Operations, Configurations, Data Definitions and Database Systems, Adaptation and Installation, Design and Implementation, Performance, Security, Safety, Resources, and Human Factors. It also includes two open categories ("Others... in-house", #15–16) intended for organisation-specific requirements. Accessibility and WCAG are not among these 14 categories. The closest category is "Human factors" (number 14), but this refers to general ergonomics and human-computer interaction, not to WCAG-defined accessibility for users with disabilities. The illustrative

example in the guide uses Performance, Portability, and Security, not accessibility. The "Others... in-house" slots are a formal extension point: the guide explicitly acknowledges that organisations may add their own NFR types beyond the standard 14. Accessibility is absent from the core list and from the examples, but the structure allows for additions.

IFPUG SNAP (Software Non-functional Assessment Process)

SNAP is IFPUG's dedicated method for sizing non-functional requirements, complementary to Function Points. FP measures WHAT the software does, SNAP measures HOW it does it (IFPUG, n.d.). SNAP is now part of the international standard ISO/IEC/IEEE 32430:2025 "Software engineering, Software non-functional size measurement" (building on IEEE 2430-2019), the most recent of the three reference points considered in this study.

SNAP recognizes four categories and fourteen subcategories of non-functional user requirements: Data Operations (covering data entry validations, logical and mathematical operations, data formatting, internal data movements, and delivering added value through data configuration), Interface Design (user interfaces, help methods, multiple input methods, and multiple output formats), Technical Environment (multiple platforms, database technology, and batch processes), and Architecture (component-based software and multiple input/output interfaces).

A full-text examination of SNAP APM Release 2.4, the current normative document, yields the following findings regarding accessibility and WCAG. The term "WCAG" appears exactly once, in the section on assessment preparation (APM v2.4, Section 1.3): "Standards and International guidelines (such as WCAG 2.1 accessibility guidelines) should be considered, if applicable to the assessed application." This is a general disclaimer, not a counting instruction:

no subcategory, no complexity parameters, and no SP calculation procedure for WCAG-related requirements is provided anywhere in the APM. Notably, APM v2.4 references WCAG 2.1 rather than the current WCAG 2.2 (W3C, 2023), reflecting that the cited accessibility standard has since evolved. The term "Accessibility" appears in APM 2.4 within the ISO/IEC 25010:2011 quality model section, listed as a sub-characteristic of Usability. A 2023 revision of ISO/IEC 25010 exists.

This study references the 2011 edition because it is the version cited in SNAP APM v2.4. A mapping table in the APM associates ISO 25010 quality characteristics to SNAP subcategories, but Accessibility has no dedicated subcategory assigned. The subcategories most relevant to accessibility requirements: 2.1 User Interfaces, 2.2 Help Methods, 2.3 Multiple Input Methods, and occasionally 1.1 Data Entry Validations, are defined in terms of general UI design and interaction methods, with no reference to WCAG success criteria, conformance levels, or disability-specific requirements. The conceptual distinction matters: While ISO 25010:2011 recognises Accessibility as a sub-characteristic of Usability and defines it with reference to users with diverse capabilities, it does not provide testable success criteria or conformance procedures. WCAG, by contrast, defines specific, binary or graded success criteria against which conformance can be verified. SNAP's measurement procedure addresses neither. Some accessibility-related requirements could potentially be associated with existing SNAP subcategories. Requirements concerning visual presentation might be counted under 2.1 User Interfaces, requirements for text alternatives could relate to 2.2 Help Methods, and keyboard navigation requirements might map to 2.3 Multiple Input Methods. The issue is not that no association is possible, but that the APM provides no explicit, reproducible procedure for making such associations. The APM provides no mapping between WCAG success criteria and SNAP subcategories. It offers no guidance on how to classify requirements that span multiple subcategories, and no counting examples involving accessibility.

WCAG organises requirements around four principles: Perceivable, Operable, Understandable, and Robust (W3C, 2023).

Each principle contains multiple success criteria that do not correspond neatly to SNAP's category structure. Without explicit APM guidance, two practitioners sizing the same set of accessibility requirements could arrive at substantially different SNAP point counts, depending on how they distribute requirements across subcategories. This is precisely what operationalisation is intended to prevent. The issue, therefore, is not simply the absence of a dedicated category, but the absence of an explicit, reproducible measurement procedure.

This study refers to that absence as non-operationalisation. SNAP is the third major software sizing framework examined in this study, after UCP TCF and the COSMIC NFR Guide. Across all three frameworks, accessibility is not operationalised as a distinct, explicitly named, and procedurally defined requirement category. This finding

is consistent with early empirical work applying SNAP in practice. Ochodek and Ozgok (2015) identified challenges in the consistent classification of non-functional requirements across SNAP subcategories, noting that practitioners may encounter difficulties in determining which subcategory applies to a given requirement. The present study finds this concern to be particularly acute for accessibility requirements, for which the APM provides no explicit classification guidance, no counting examples, and no mapping to WCAG success criteria. This is corroborated by Mansoor and Ochodek (2016), who developed a semi-automated tool for measuring SNAP subcategory 2.1 in web applications. The subcategory definition explicitly lists accessibility alongside usability and attractiveness as non-functional characteristics it is intended to address. Yet the proposed measurement approach operates entirely on visual UI properties derived from HTML and CSS (button captions, background colours, font sizes, and margins) and does not include any accessibility-specific criteria, WCAG success criteria, or assistive technology compatibility checks. Even in this technically advanced application of subcategory 2.1, accessibility remains unmeasured.

The definition names it, but the proposed measurement procedure does not explicitly operationalise it. This is precisely the pattern this study identifies across all three frameworks. A potential counter-argument is that SNAP intentionally sizes non-functional requirements through generic implementation categories rather than domain-specific requirement types. On this view, accessibility does not need its own category any more than GDPR compliance, localisation, or ESG reporting need dedicated SNAP subcategories. This is a reasonable position, but it does not apply to accessibility without qualification. The reason is that accessibility is not only a domain-specific concern: it is explicitly named as a quality characteristic within ISO/IEC 25010:2011, the quality model on which SNAP's own subcategory structure is built. The APM includes a mapping table linking ISO 25010 quality characteristics to SNAP subcategories. Accessibility appears in this table as a quality characteristic. It has no SNAP subcategory assigned to it. This creates a specific tension: the standard that SNAP uses as its quality reference model recognises accessibility, but the measurement method does not operationalise it. The absence is therefore not a neutral gap in coverage, but a measurable gap between SNAP's quality reference model and its operational measurement procedure. This is a stronger finding than simply noting that accessibility is missing from the framework.

Synthesis: The Accessibility Gap

The three frameworks examined, Use Case Points (UCP), COSMIC Function Points, and IFPUG SNAP, represent distinct approaches to software sizing developed across three decades. Despite their differences in method and scope, they share a common structural gap with respect to accessibility as a non-functional requirement. Table 2 summarises the findings across the three analytical questions introduced in Section 1.

Table 2 Accessibility gap across three major software sizing frameworks.

Sizing Framework	Accessibility/WCAG explicitly named?	Operationalised? (counting rules, calculation)	Structural extension point?
UCP (Kaner, 1993)	No, T7 "Easy to Use" is closest (general usability only)	No	No formally defined extension mechanism, the factor list is not explicitly closed to additions, but no validated procedure for adding factors exists
COSMIC NFR Guide (Abran et al., 2020)	No, "Human factors" (#14) is closest. Accessibility absent from 14 ISO/IEEE/ECSS categories	No	Yes, formal "Others... in-house" slots (#15-16) explicitly provided
IFPUG SNAP APM v2.4 (ISO/IEC/IEEE 32430:2025)	Partial, WCAG appears once as general disclaimer. Accessibility listed as ISO 25010 Usability sub-characteristic without dedicated SNAP subcategory	No, no subcategory, no complexity parameters, no SP calculation for accessibility	Implicit, ISO 25010 mapping table has Accessibility without assigned SNAP subcategory

Note: "Operationalised" means a framework provides counting rules, complexity parameters, and a calculation procedure sufficient to produce a numeric size estimate for accessibility-related requirements. Presence of the word "accessibility" or "WCAG" without such procedures does not constitute operationalisation.

The most significant finding in Table 2 concerns SNAP. Unlike UCP and COSMIC, where accessibility is simply absent, SNAP occupies a more nuanced position: the term "accessibility" appears in the APM 2.4 as a sub-characteristic of Usability within the ISO/IEC 25010:2011 quality model, and the term "WCAG" appears once as an example of a standard that "should be considered, if applicable." Neither appearance results in operationalisation: the APM provides no dedicated subcategory, no complexity parameters, and no SP calculation procedure for accessibility requirements.

This distinction matters: the standard's own quality reference model recognises accessibility, but the measurement method built on that model does not translate that recognition into measurable, countable units.

Accessibility is named but not sized. Across the three frameworks examined in this study, the structural extension points identified in Table 2, namely UCP's TCF list (extension not formally documented), COSMIC's "Others... in-house" slots, and SNAP's ISO 25010 mapping without assigned subcategories, demonstrate that the frameworks are not conceptually closed to new NFR categories. The absence of accessibility is therefore not a structural impossibility, but a gap in the current scope of each method. This gap has a practical consequence: cost estimates for software projects subject to accessibility requirements may omit the additional effort required for WCAG conformance. Where accessibility requirements are not represented in the sizing model, they cannot be estimated within the framework, and where they cannot be estimated, they tend to be treated as an afterthought, addressed through costly retrofitting rather than designed in from the outset.

The economic dimension of this gap is well-established in the broader accessibility literature. Wentz, Jaeger & Lazar (2011) argue that the cost of building accessibility into new development is minimal compared to the cost of retrofitting years later, and that retrofitted accessibility solutions tend to be qualitatively inferior, described as "separate but not equal", and neglected over time (Wentz et al., 2011; Wentz et al., 2014). When established sizing methods do not include accessibility as a recognised requirement category, this may contribute to systematic underestimation for any project where WCAG conformance is a requirement. This is consistent with a recent systematic literature review of AI applications in functional size measurement (Czarnacka-Chrobot, 2025), which identifies integration of non-functional requirements into FSM models as an open research direction and lists usability, reliability, performance, and security among relevant NFR types, without mentioning accessibility. Accessibility remains outside the standard NFR enumeration across both sizing frameworks and recent FSM research.

Taken together, the findings point to a consistent and triangulated gap: three independently developed, established software sizing frameworks, covering both functional complexity adjustment (UCP TCF), non-functional sizing guidance (COSMIC NFR Guide), and dedicated NFR measurement (IFPUG SNAP), none operationalise accessibility as a distinct, measurable requirement category. The findings suggest that the gap may be structural rather than an isolated omission in any single framework, and its practical consequences for software cost estimation in accessibility-regulated contexts are direct.

Discussion and Implications

Implications for Software Cost Estimation

The most direct implication of the findings concerns software cost estimation. When a sizing method does not include accessibility as a recognised requirement category, the effort needed to meet accessibility requirements is not captured in the size estimate. A practitioner applying UCP, COSMIC, or SNAP according to their respective normative documents will produce an estimate that does not reflect the effort required to analyse, implement, test, and verify WCAG conformance. Any adjustment for accessibility must be made informally, outside the formal method.

Informal adjustments have a known cost: they reduce the comparability, auditability, and reproducibility of estimates.

Two organisations estimating the same project using the same sizing method may arrive at substantially different totals if one applies an informal accessibility correction and the other does not. This undermines one of the primary benefits of standardised sizing, the ability to produce consistent, comparable estimates across projects, teams, and organisations.

The broader consequence is that accessibility effort tends to be underrepresented in project budgets and schedules, contributing to its treatment as a retrofit activity rather than a designed-in requirement. As Wentz, Jaeger and Lazar (2011) have shown, retrofitting is more costly and produces qualitatively inferior results.

The gap identified in this study has particular relevance in procurement contexts where accessibility is a contractual or regulatory requirement. In public sector IT procurement, functional size measurement methods are widely used as the basis for project valuation, vendor selection, and contract pricing. Where accessibility requirements are present, whether by regulation, technical specification, or organisational policy, and the sizing

method used does not account for them, the resulting estimate risks excluding a component of the required work. This creates a risk that accessibility compliance is either omitted from the scope of the contract, underfunded within it, or left to be addressed after delivery at higher cost. This risk is not hypothetical. Empirical research on accessibility in public sector web services consistently shows that formal compliance requirements do not automatically translate into accessible products (see companion studies: Maciejewska, 2026a; Maciejewska, 2026b). One possible contributing factor, consistent with the gap identified in this study, is that the cost of accessibility compliance may not be adequately represented in the estimation models used to plan and price development work. Where sizing methods treat accessibility as invisible, procurement processes will tend to treat it as an afterthought.

Implications for FSM Research

The consistency of the gap across three independently developed frameworks, UCP, COSMIC, and SNAP, suggests that accessibility has not historically been considered a distinct requirement category in the design of software sizing methods. This is consistent with a recent systematic literature review of AI applications in functional size measurement (Czarnacka-Chrobot, 2025), which identifies integration of non-functional requirements into FSM models as an open research direction and lists usability, reliability, performance, and security among relevant NFR types, without mentioning accessibility. The omission across both frameworks and research literature suggests a shared blind spot rather than an isolated oversight in any single method. For FSM research, the structural extension points identified in this study, COSMIC's open "Others... in-house" NFR slots, and SNAP's ISO 25010 mapping table with unassigned subcategories suggest that extending existing frameworks to include accessibility does not require fundamental redesign. An important direction for future research is to define what such an extension would look like in practice: what requirement types it would cover, what complexity parameters it would use, and how it would be calibrated using empirical project data. A related empirical question is whether the number or severity of accessibility violations detectable through automated tools correlates with the sizing effort required to remediate them, which would provide an evidence base for any proposed extension.

Both directions are outside the scope of this study. The value of such an extension is supported by recent empirical work on analogous NFR types. Daud and Malik (2024) demonstrated that incorporating security-based NFR into early software size estimation models improves estimation accuracy, using SNAP subcategory 1.1 (Data Entry Validations) to quantify security requirements and building a validated SLOC estimation model from an industrial dataset. Their findings suggest that when NFR are operationalised within a sizing framework, they can be systematically included in estimation, reducing the gap between projected and actual effort.

Accessibility requirements, which generate specific and verifiable implementation effort across multiple development phases, present an analogous case: the absence of an operationalisation mechanism in current frameworks is not a theoretical concern but a practical obstacle to accurate cost estimation. A key difference, however, is that security validations in the sense addressed by Daud and Malik map naturally to a single SNAP subcategory (1.1). WCAG requirements span multiple subcategories simultaneously, covering interface design, help methods, input methods, and data validation, without any explicit classification guidance in the APM. This is precisely why a dedicated operationalisation procedure is needed rather than a simple mapping to existing constructs. This study has limitations.

It examines three frameworks based on their normative documents and does not cover all sizing methods in use. It does not examine how practitioners handle accessibility requirements in real projects, which would require empirical data. The conclusions apply to the frameworks as documented, not to how they may be applied in practice. Importantly, this study identifies the absence of an operationalisation mechanism but does not quantify the magnitude of any resulting underestimation. Whether the gap translates into material cost differences in real projects is an empirical question that lies outside the scope of this analysis.

Conclusion

This study examined three established software sizing frameworks, Use Case Points (UCP), COSMIC Function Points, and IFPUG SNAP, to determine whether accessibility is operationalised as a non-functional requirement category in any of them. The answer is consistent across all three frameworks examined in this study: it is not. None of the frameworks provides an explicit and reproducible procedure for sizing accessibility requirements. Accessibility is named but not sized. Two of the three frameworks have formal structural extension points, COSMIC's open "Others... in-house" slots and SNAP's ISO 25010 mapping table with unassigned subcategories.

UCP's factor list has no formally documented extension mechanism, but is not explicitly closed to additions either. In each of the three frameworks, the absence of accessibility is not a fundamental structural limitation. It is a gap in the current scope of each method. Closing this gap would require defining accessibility as a measurable cost driver with its own counting rules and sizing procedure. The practical consequence is a risk of systematic underestimation of accessibility costs in software projects. When accessibility effort is not captured by the sizing model, it tends to be treated as an afterthought. Future work should explore how accessibility compliance could be operationalised within existing frameworks, and whether empirical project data can support the calibration of such an extension.

References

- Abran, A., Al-Sarayreh, K., and Lesterhuis, A. (Eds.) (2020) *Non-Functional Requirements and COSMIC Sizing: Practitioner's Guide* (v1.0, minor update December 2020). COSMIC.
- <https://cosmic-sizing.org/wp-content/uploads/2020/05/System-NFR-practitioners-v1.0-December-2020.pdf>
- Buglione, L. and Di Cola, F. (2025) 'IFPUG SNAP and its ISBSG Data Collection Questionnaire.' [Conference presentation]. 2° *EventoMetrico 2025 GUFPI-ISMA & IT Confidence 2025 ISBSG*, Florence, Italy and online, September 19, 2025. Non-Functional Sizing Standards Committee (NFSSC), IFPUG.
- Czarnacka-Chrobot, B. (2013a) 'Rationalization of Business Software Systems Development and Enhancement Projects Investment Decisions on the Basis of Functional Size Measurement.' *International Journal of Software Engineering and Knowledge Engineering*, 23(6), 839–868. <https://doi.org/10.1142/S0218194013500228>
- Czarnacka-Chrobot, B. (2013b) 'Analysis of Software Development and Enhancement Projects Work Effort per Unit based on the COSMIC Method with regard to Technological Factors.' *Journal of Software Engineering and Applications*, 6(11), 597–609. <https://doi.org/10.4236/jsea.2013.611072>
- Czarnacka-Chrobot, B. (2025) 'AI applications in software functional size measurement: A systematic literature review.' *The Journal of Systems and Software*, 232, 112686. Advance online publication. <https://doi.org/10.1016/j.jss.2025.112686>
- Daud, M. and Malik, A. A. (2024) 'Exploring the impact of security-based non-functional requirements on early software size estimation.' *Computing and Informatics*, 43(5), 1234–1255. https://doi.org/10.31577/cai_2024_5_1234
- European Telecommunications Standards Institute (ETSI). (2021) *EN 301 549 v3.2.1: Accessibility requirements for ICT products and services*. ETSI.
- https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf
- IFPUG. (2015/2025) *SNAP Assessment Practices Manual (APM) v2.4*. International Function Point Users Group. <https://ifpug.org/ifpug-standards/snap> [No formal DOI; v2.3 Quick Reference Guide publicly available; APM v2.4 referenced in Buglione and Di Cola (2025) and ISO/IEC/IEEE 32430:2025]
- IFPUG. (n.d.) *Software Non-functional Assessment Process (SNAP)*. International Function Point Users Group. <https://ifpug.org/ifpug-standards/snap> [Accessed 5 July 2026]
- International Organization for Standardization (ISO). (2011) *ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*.
- International Organization for Standardization, International Electrotechnical Commission, and Institute of Electrical and Electronics Engineers. (2025) *ISO/IEC/IEEE 32430:2025 - Software engineering, Software non-functional size measurement*.
- Karner, G. (1993) *Metrics for Objectory* (Diploma thesis No. LiTH-IDA-Ex-9344:21). University of Linköping, Sweden.
- Maciejewska, K. (2026a) 'Consistency of Browser-Based Accessibility Evaluation Tools: An Empirical Study Across EU Healthcare Websites.' *Proceedings of the 47th International Business Information Management Association (IBIMA)*, ISSN: 2767-9640, Madrid, Spain. [in press]

- Maciejewska, K. (2026b) 'Accessibility Declarations vs. Automated Audit Results: A Cross-Country Study of Government Websites in Six EU Member States.' *Proceedings of the 47th International Business Information Management Association (IBIMA)*, ISSN: 2767-9640, Madrid, Spain. [in press]
- Mansoor, H. and Ochodek, M. (2016) 'Towards Semi-automatic Size Measurement of User Interfaces in Web Applications with IFPUG SNAP.' In *Proceedings of the 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-Mensura)*, pp. 191–194. IEEE. <https://doi.org/10.1109/IWSM-Mensura.2016.7>
- Ochodek, M., Nawrocki, J. and Kwarciak, K. (2011) 'Simplifying effort estimation based on Use Case Points.' *Information and Software Technology*, 53(3), 200–213. <https://doi.org/10.1016/j.infsof.2010.10.005>
- Ochodek, M. and Ozgok, B. (2015) 'Functional and Non-functional Size Measurement with IFPUG FPA and SNAP, Case Study.' In R. Silhavy et al. (Eds.), *Software Engineering in Intelligent Systems. Advances in Intelligent Systems and Computing*, vol. 349. Springer. https://doi.org/10.1007/978-3-319-18473-9_3
- Wentz, B., Jaeger, P. T. and Lazar, J. (2011) 'Retrofitting accessibility: The legal inequality of after-the-fact online access for persons with disabilities in the United States.' *First Monday*, 16(11). <https://doi.org/10.5210/fm.v16i11.3666>
- Wentz, B., Jaeger, P. T. and Lazar, J. (2014) 'Accessibility in Practice: A Process-Driven Approach to Accessibility.' *Universal Usability: Designing Computer Interfaces for Diverse User Populations*. Wiley.
- https://doi.org/10.1007/978-3-319-05095-9_10
- World Wide Web Consortium (W3C). (2023) *Web Content Accessibility Guidelines (WCAG) 2.2*. W3C Recommendation. <https://www.w3.org/TR/WCAG22/>