Юбилейна международна научна конференция „Синергетика и рефлексия в обучението по математика“, 22-24 октомври 2025 г., Пампорово, България
Anniversary International Scientific Conference “Synergetics and Reflection in Mathematics Education”, October 22-24, 2025, Pamporovo, Bulgaria

INTEGRATING RELIABILITY MODELING INTO VERIFICATION AND VALIDATION EDUCATION

Maria Vasileva[1,*], Georgi Penchev[2]

[1, 2] *Faculty of Mathematics and Informatics, Paisii Hilendarski University of Plovdiv, 24 Tzar Asen, 4000 Plovdiv, Bulgaria*

[1,*] *Corresponding author: mariavasileva@uni-plovdiv.bg*

[2] *georgi.penchev@uni-plovdiv.bg*

Abstract. This paper presents a university-level course in Software Verification and Validation (V&V), with emphasis on Software Reliability Growth Models (SRGM) as a core tool for quantitative reliability analysis. Aimed at students in Software Engineering and Computer Science, the course combines theory, laboratory work, and a project component. It aligns with international standards, including ABET, CS2023, and SWEBOK v4.0, and equips students with essential skills in formal methods, statistical modeling, and software quality evaluation. A comparative overview of similar university courses highlights the distinctiveness of integrating SRGM into the core of V&V education.

Key Words: *software verification and validation, software reliability, software reliability growth models, educational standards.*

Introduction

Courses in software verification and validation (V&V) are fundamental in preparing future software engineers and computer scientists, as they equip students with theoretical and practical tools to ensure software quality. Verification and validation are defined as the processes of investigating whether a software system satisfies formal specifications and standards and fulfills its intended purpose. Verification examines whether the software is built right, in conformity with requirements, while validation assesses whether the right product has been built to meet user needs. Both processes are critical for the development of robust, dependable, and user-oriented software systems.

In software verification and validation, systems are analyzed not only through empirical testing but also through formal, mathematically grounded methods that ensure compliance with explicit specifications. A well-structured V&V curriculum combines theoretical foundations with practical techniques such as formal verification, model checking, systematic testing strategies, and the use of software quality metrics. The focus extends beyond defect detection, aiming to provide substantiated evidence that the software behaves correctly and predictably under defined conditions.

A critical component of this process is modeling software reliability [1, 2], which supports the prediction of failure behavior, guides resource allocation during development, and strengthens confidence in system dependability. Reliability is not merely a desirable characteristic but is recognized as a core quality attribute that encompasses aspects such as fault tolerance, recoverability, and availability that are essential for ensuring sustained performance in real-world environments. Reliability is one of the core attributes defined in the ISO/IEC 25010:2023 standard (*Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model*). It is defined as the degree to which a system or component performs specified functions under given conditions for a defined period of time without failure.

In contemporary software engineering, testing has become increasingly automated and is supported by artificial intelligence. At the same time, modern software systems often incorporate components from open-source software (OSS) projects. The development of these projects is decentralized and dynamic, which creates specific challenges for ensuring reliability. Therefore, academic programs in Verification and Validation (V&V) should place a stronger emphasis on quantitative approaches to reliability assessment. When empirical test data is scarce, as is often the case in OSS projects, *Software Reliability Growth Models* (SRGM) offer an effective way to forecast fault trends and failure behavior. These models are critical for making decisions about software release readiness and the need for further testing. The diversity of SRGMs in the literature makes them a valuable part of V&V education and research.

Furthermore, the use of quantitative models such as SRGM equips students with analytical tools for evaluating reliability based on observed failure data. This approach bridges the gap between theoretical knowledge and real-world practice, reinforcing the importance of mathematics and statistics in software quality assurance. By embedding these elements into the V&V course structure, students are better prepared to address the complexities of professional software development, where accountability, correctness, and measurable quality are essential requirements rather than optional goals.

Reliability modeling, grounded in probability theory, statistical inference, and approximation methods, illustrates the reflective application of mathematics in computer science. In this context, mathematics is not regarded solely as abstract discipline, but as a methodological instrument for understanding and optimizing complex technological systems.

The main aim of the presented paper is to present a curriculum framework for educating future software engineers and computer scientists in the field of software verification and validation (V&V), with a particular focus on reliability analysis. Upon completion of the course, students will acquire the following competencies:

- Ability to distinguish and apply the principles of software verification and validation;
- Proficiency in implementing *Software Reliability Growth Models (SRGM)* for testing and quality evaluation;
- Skills in using statistical tools and visualization techniques to model and assess software reliability;
- Capability to conduct quantitative analyses and formulate data-driven recommendations regarding testing completion and release readiness.

Alignment with Educational Standards

The ACM/IEEE Computer Science Curricula 2023 (CS2023) [3] knowledge model comprises 17 Knowledge Areas (KAs). Compared to the 2013 edition, there is a notable increase in required core hours: from 41 hours in CS2013 (Tier-1: 37, Tier-2: 4) to 200 core hours in CS2023 (CS Core: 55 and KA Core: 145). Topics within the *Mathematical and Statistical Foundations (MSF)* Knowledge Area are classified as KA Core, as they are prerequisites for core topics across several other Knowledge Areas. By integrating topics from the MSF Knowledge Area, such as statistical inference, probabilistic reasoning, and model evaluation, the proposed course addresses core competencies that are now recognized as foundational across multiple domains.

Furthermore, the curriculum directly supports three key Knowledge Units (KUs) within the Software Engineering (SE) Knowledge Area:

- *SE-Validation: Software Verification and Validation* is addressed through systematic testing strategies, model-based validation, and quantitative evaluation of test effectiveness using SRGMs;
- *SE-Reliability: Software Reliability* is a central theme, as students analyze defect trends, understand failure modes, and explore models that simulate software reliability growth during testing;

- *SE-Formal: Formal Methods* is supported through the application of mathematical models and estimation techniques, which draw on core material from discrete mathematics, logic, and formal reasoning.

The ABET Engineering Accreditation Commission outlines clear expectations for academic programs in Computer Science and Software Engineering. According to the 2025–2026 Criteria [4] for Accrediting Computing Programs the curriculum must include fundamentals of programming, software design and construction, requirements analysis, security, verification and validation, as well as mathematical subjects such as discrete mathematics, probability, and statistics with applications appropriate to software engineering. Students must acquire competencies in *the application of verification and validation techniques to ensure that software meets its specified requirements and operates reliably*. These requirements justify the integration of Software Reliability Growth Models into the curriculum. SRGMs offer a structured, statistical approach to evaluating software reliability based on failure data, illustrating how quantitative analysis supports engineering decisions related to software quality and dependability. Their inclusion strengthens students' ability to apply formal verification techniques and assess system behavior in realistic conditions, both of which are central to ABET's criteria for high-quality computing education. Also foster critical thinking, risk evaluation, and evidence-based assurance of software correctness.

In addition, the course is consistent with the structure of the *Software Engineering Body of Knowledge (SWEBOK Guide)* [5], published by the IEEE Computer Society. SWEBOK identifies 18 knowledge areas (KAs) that reflect widely accepted practices in both industry and academia. Among them, the proposed curriculum addresses the following particularly well:

- *Software Testing* – through test evaluation metrics, SRGM-based failure prediction, and defect modelling;
- *Software Quality* – by introducing statistical measures of reliability and supporting continuous quality assessment;
- *Mathematical Foundations* – via formal methods and probabilistic reasoning essential for understanding model behavior;
- *Engineering Foundations* – by linking theory with data-driven decision-making in the software development life cycle.

The European e-Competence Framework for ICT professionals [6] have been formally published as a European Standard by CEN, the European Committee for Standardization. It gives organizations, business, and ICT practitioners in Europe a common language to describe digital competencies and proficiency levels. It is structured from 5 e-Competence areas, derived from the ICT business processes PLAN – BUILD – RUN – ENABLE – MANAGE. A set of reference e-Competences for each area, with a generic description for each competence. 40 competences identified in total provide the European generic reference definitions of the e-CF 3.0. Proficiency levels of each e-Competence provide European reference level specifications on e-Competence levels e-1 to e-5, which are related to the EQF levels 3 to 8. Table 1 illustrates how specific competencies from the European e-Competence Framework (e-CF) are addressed and developed through activities within the course on software verification, validation, and reliability analysis.

Table 1. Alignment of Course Activities with Selected e-CF Competences and Levels

e-CF areas	e-Competences identified	V & V course	
		e-Competence proficiency level	Application in the Course
A. PLAN	A.6 Application Design	e-3	Selecting and justifying an appropriate SRGM for a given software system
B. BUILD	B.2 Component Integration	e-3	Implementing reliability models using existing software failure data
C. RUN	C.4 Problem Management	e-2	Analyzing defect logs and predicting future failures
D. ENABLE	D.10 Information and Knowledge Management	e-3	Processing and analyzing reliability data to support informed decisions
E. MANAGE	E.3 Risk Management	e-3	Evaluating when to stop testing based on quantitative risk analysis
E. MANAGE	E.4 Relationship Management	e-3	Reporting and communicating reliability findings to the development team

Comparative Analysis with Other University Courses

The proposed curriculum framework is intended for students in Software Engineering and Computer Science at the Faculty of Mathematics and Informatics, Paisii Hilendarski University of Plovdiv. While several leading academic institutions have addressed related topics, SRGM are seldom the central focus of such courses. In this context, it is important to acknowledge the contribution of the Bulgarian school in this field [7-11]. Table 2 provides a comparative analysis of university programs that integrate software reliability into their curricula.

Table 2. Comparative Analysis of University Programs Incorporating Software Reliability into Their Curricula

University	Course Title	Description
University of Bolzano (Italy)	Software Reliability and Testing *(MSc)*	Covers software verification and testing with a focus on reliability, including stochastic processes, Poisson processes, and SRGMs.
University of Bolzano (Italy)	Verification and Reliability of Dependable Systems *(MSc)*	Combines formal verification techniques (e.g., model checking) with statistical reliability modeling, including SRGMs and real-world implementation case studies.
University of Technology Sydney (Australia)	Software Reliability Growth Modeling (Seminar)	One-day academic seminar covering multiple SRGMs, their classification, statistical methods, and domain-specific adaptations.
University of California, Los Angeles (USA)	Software Reliability (Course or Seminar Series)	Includes SRGMs, model checking, and techniques for formal and statistical verification of software reliability.
University of Memphis, FedEx Institute of Technology (USA)	Automated, Cloud-based and Real-Time SRGM (Two-Day Workshop)	Focuses on automated SRGM modeling in agile software environments using daily failure data and cloud-based tools.

Conclusion

The proposed course on Software Verification and Validation, with a focus on SRGM, addresses a critical gap in current computing curricula by introducing formal, data-driven methods for evaluating software reliability. By aligning with the competencies defined in CS2023, SWEBOK v4.0, and ABET accreditation criteria, the course not only meets international educational standards but also prepares students for real-world software quality challenges. Comparative analysis with other academic programs confirms the relevance and distinctiveness of this curriculum framework. The integration of SRGMs not only enhances students’ analytical capacity and equips them to confront real-world software quality challenges but also fosters the development of dependable and ethically responsible systems. Importantly, the study demonstrates that mathematics should not be regarded solely as an abstract discipline, but as a methodological instrument for understanding and optimizing complex technological systems. This interdisciplinary approach reinforces the synergetic relationship between theory and practice, cultivating reflective, analytical, and ethical competencies that are essential for modern software engineering education.

Acknowledgments

This research was funded by project No. FP25-FMI-010, “Innovative interdisciplinary research in Informatics, Mathematics, and Pedagogy of Education”, of the Scientific Fund of the Paisii Hilendarski University of Plovdiv, Bulgaria.

References

[1] Yamada, S., *Software reliability modeling: fundamentals and applications*, Vol. 5, Tokyo: Springer, 2014, ISBN: 978-4-431-54564-4

[2] Yamada, S., Y. Tamura, *OSS Reliability Measurement and Assessment*, Springer Series in Reliability Engineering, Springer: Cham, Switzerland, 2016, ISBN: 978-3-319-31818-9

[3] ACM/IEEE Computer Science Curricula 2023 https://ieeecs-media.computer.org/media/education/reports/CS2023.pdf (Accessed August 27, 2025)

[4] ABET Engineering Accreditation Commission https://www.abet.org/wp-content/uploads/2024/11/2025-2026_EAC_Criteria.pdf (Accessed August 27, 2025)

[5] Washizaki, H. eds., *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*, *Version 4.0*, IEEE Computer Society, 2024

[6] European e-Competence Framework 3.0 https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/european-e-competence-framework-e-cf (Accessed August 27, 2025)

[7] Pavlov, N., A. Iliev, A. Rahnev, N. Kyurkchiev, *Some software reliability models: approximation and modeling aspects*, LAP LAMBERT Academic Publishing, 2018, ISBN: 978-613-9-82805-0

[8] Pavlov, N., A. Iliev, A. Rahnev, N. Kyurkchiev, *Nontrivial models in debugging theory (Part 2)*, LAP LAMBERT Academic Publishing, 2018, ISBN: 978-613-9-87794-2

[9] Rahneva, O, A. Golev, G. Spasov, *Investigations on some new models in debugging and growth theory (Part 3)*, LAP LAMBERT Academic Publishing, 2020, ISBN: 978-620-66655-8

[10] Kyurkchiev, N., A. Iliev, A. Golev, A. Rahnev, *Some non-standard models in "Debugging and test theory" (Part 4),* Plovdiv University Press: Plovdiv, Bulgaria, 2020, ISBN: 978-619-2-02584-7

[11] Kyurkchiev, V., A. Iliev, A. Rahnev, T. Terzieva, E. Angelova, On Some Understudied Models with Applications in the Field of Debugging Theory, *Communications in Applied Analysis*, 2022, 26 (1), 9–18