

1. Record Nr.	UNINA9910591037803321
Titolo	Handbook on artificial intelligence-empowered applied software engineering . Volume 1 Novel methodologies to engineering smart software systems // Maria Virvou [and three others] editors
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-031-08202-8
Descrizione fisica	1 online resource (342 pages)
Collana	Artificial intelligence-enhanced software and systems engineering ; ; Volume 2
Disciplina	006.3
Soggetti	Artificial intelligence - Industrial applications Software engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Intro -- Foreword -- Preface -- Contents -- 1 Introduction to Handbook on Artificial Intelligence-Empowered Applied Software Engineering-VOL.1: Novel Methodologies to Engineering Smart Software Systems -- 1.1 Editorial Note -- 1.2 Book Summary and Future Volumes -- Bibliography for Further Reading -- Part I Survey of Recent Relevant Literature -- 2 Synergies Between Artificial Intelligence and Software Engineering: Evolution and Trends -- 2.1 Introduction -- 2.2 Methodology -- 2.3 The Evolution of AI in Software Engineering -- 2.4 Top Authors and Topics -- 2.5 Trends in AI Applications to Software Engineering -- 2.5.1 Machine Learning and Data Mining -- 2.5.2 Knowledge Representation and Reasoning -- 2.5.3 Search and Optimisation -- 2.5.4 Communication and Perception -- 2.5.5 Cross-Disciplinary Topics -- 2.6 AI-Based Tools -- 2.7 Conclusion -- References -- Part II Artificial Intelligence-Assisted Software Development -- 3 Towards Software Co-Engineering by AI and Developers -- 3.1 Introduction -- 3.2 Software Development Support and Automation Level by Machine Learning -- 3.2.1 Project Planning: Team Composition -- 3.2.2 Requirements Engineering: Data-Driven Persona -- 3.2.3 Design: Detection of Design Patterns -- 3.2.4 Categorization of Initiative and Level of Automation -- 3.3 Quality of AI

Application Systems and Software -- 3.3.1 Metamorphic Testing --
3.3.2 Improving Explainability -- 3.3.3 Systems and Software
Architecture -- 3.3.4 Integration of Goals, Strategies, and Data -- 3.4
Towards Software Co-Engineering by AI and Developers -- 3.5
Conclusion -- References -- 4 Generalizing Software Defect Estimation
Using Size and Two Interaction Variables -- 4.1 Introduction -- 4.2
Background -- 4.3 A Proposed Approach -- 4.3.1 Selection of Sample
Projects -- 4.3.2 Data Collection -- 4.3.3 The Scope and Decision
to Go with 'Interaction' Variables.
4.3.4 Data Analysis and Results Discussion -- 4.3.5 The Turning Point
-- 4.3.6 Models Performance-Outside Sample -- 4.4 Conclusion
and Limitations -- 4.5 Future Research Directions -- 4.6 Annexure-
Model Work/Details -- References -- 5 Building of an Application
Reviews Classifier by BERT and Its Evaluation -- 5.1 Background -- 5.2
The Process of Building a Machine Learning Model -- 5.3 Dataset --
5.4 Preprocessing -- 5.5 Feature Engineering -- 5.5.1 Bag of Words
(BoW) [4] -- 5.5.2 FastText [5, 6] -- 5.5.3 Bidirectional Encoder
Representations from Transformers (BERT) Embedding [7] -- 5.6
Machine-Learning Algorithms -- 5.6.1 Naive Bayes -- 5.6.2 Logistic
Regression -- 5.6.3 BERT -- 5.7 Training and Evaluation Methods --
5.8 Results -- 5.9 Discussion -- 5.9.1 Comparison of Classifier
Performances -- 5.9.2 Performance of the Naive Bayes Classifiers --
5.9.3 Performance of the Logistic Regression Classifiers -- 5.9.4
Visualization of Classifier Attention Using the BERT -- 5.10 Threats
to Validity -- 5.10.1 Labeling Dataset -- 5.10.2 Parameter Tuning --
5.11 Summary -- References -- 6 Harmony Search-Enhanced Software
Architecture Reconstruction -- 6.1 Introduction -- 6.2 Related Work --
6.3 HS Enhanced SAR -- 6.3.1 SAR Problem -- 6.3.2 HS Algorithm --
6.3.3 Proposed Approach -- 6.4 Experimentation -- 6.4.1 Test
Problems -- 6.4.2 Competitor approaches -- 6.5 Results
and Discussion -- 6.6 Conclusion and Future Work -- References -- 7
Enterprise Architecture-Based Project Model for AI Service System
Development -- 7.1 Introduction -- 7.2 Related Work -- 7.3 AI Service
System and Enterprise Architecture -- 7.3.1 AI Service System -- 7.3.2
Enterprise Architecture and AI Service System -- 7.4 Modeling Business
IT Alignment for AI Service System -- 7.4.1 Generic Business-AI
Alignment Model -- 7.4.2 Comparison with Project Canvas Model.
7.5 Business Analysis Method for Constructing Domain Specific
Business-AI Alignment Model -- 7.5.1 Business Analysis Tables --
7.5.2 Model Construction Method -- 7.6 Practice -- 7.6.1 Subject
Project -- 7.6.2 Result -- 7.7 Discussion -- 7.8 Conclusion --
References -- Part III Software Engineering Tools to Develop Artificial
Intelligence Applications -- 8 Requirements Engineering Processes
for Multi-agent Systems -- 8.1 Introduction -- 8.2 Background --
8.2.1 Agents, Multiagent Systems, and the BDI Model -- 8.2.2
Requirements Engineering -- 8.3 Techniques and Process
of Requirements Engineering for Multiagent Systems -- 8.3.1 Elicitation
Requirements Techniques for Multiagent Systems -- 8.3.2
Requirements Engineering Processes for Multiagent Systems -- 8.3.3
Requirements Validation -- 8.4 Conclusion -- References -- 9 Specific
UML-Derived Languages for Modeling Multi-agent Systems -- 9.1
Introduction -- 9.2 Background -- 9.2.1 UML -- 9.2.2 Agents,
Multiagent Systems, and the BDI Model -- 9.2.3 BDI Models -- 9.3
AUML-Agent UML -- 9.4 AORML-Agent-Object-Relationship Modeling
Language -- 9.4.1 Considerations About AORML -- 9.5 AML-Agent
Modeling Language -- 9.5.1 Considerations About AML -- 9.6 MAS-
ML-Multiagent System Modeling Language -- 9.6.1 Considerations
About MAS-ML -- 9.7 SEA-ML-Semantic Web Enabled Agent Modeling

Language -- 9.7.1 Considerations -- 9.8 MASRML-A Domain-Specific Modeling Language for Multi-agent Systems Requirements -- 9.8.1 Considerations -- References -- 10 Methods for Ensuring the Overall Safety of Machine Learning Systems -- 10.1 Introduction -- 10.2 Related Work -- 10.2.1 Safety of Machine Learning Systems -- 10.2.2 Conventional Safety Model -- 10.2.3 STAMP and Its Related Methods -- 10.2.4 Standards for Software Lifecycle Processes and System Lifecycle Processes -- 10.2.5 Social Technology Systems and Software Engineering.

10.2.6 Software Layer Architecture -- 10.2.7 Assurance Case -- 10.2.8 Autonomous Driving -- 10.3 Safety Issues in Machine Learning Systems -- 10.3.1 Eleven Reasons Why We Cannot Release Autonomous Driving Cars -- 10.3.2 Elicitation Method -- 10.3.3 Eleven Problems on Safety Assessment for Autonomous Driving Car Products -- 10.3.4 Validity to Threats -- 10.3.5 Safety Issues of Automatic Operation -- 10.3.6 Task Classification -- 10.3.7 Unclear Assurance Scope -- 10.3.8 Safety Assurance of the Entire System -- 10.3.9 Machine Learning and Systems -- 10.4 STAMP S&S -- S Method -- 10.4.1 Significance of Layered Modeling of Complex Systems -- 10.4.2 STAMP S&S -- S and Five Layers -- 10.4.3 Scenario -- 10.4.4 Specification and Standard -- 10.5 CC-Case -- 10.5.1 Definition of CC-Case -- 10.5.2 Technical Elements of CC-Case -- 10.6 Measures for Autonomous Driving -- 10.6.1 Relationship Between Issues and Measures Shown in This Section -- 10.6.2 Measure 1: Analyze Various Quality Attributes in Control Action Units -- 10.6.3 Measure 2: Modeling the Entire System -- 10.6.4 Measure 3: Scenario Analysis and Specification -- 10.6.5 Measure 4: Socio-Technical System -- 10.7 Considerations in Level 3 Autonomous Driving -- 10.7.1 Example of Autonomous Driving with the 5-layered Model of STAMP S&S -- S -- 10.8 Conclusion -- References -- 11 MEAU: A Method for the Evaluation of the Artificial Unintelligence -- 11.1 Introduction -- 11.2 Machine Learning and Online Unintelligence: Improvisation or Programming? -- 11.3 The New Paradigm of Information from Digital Media and Social Networks -- 11.4 Numbers, Images and Texts: Sources of Errors, Misinformation and Unintelligence -- 11.5 MEAU: A Method for the Evaluation of the Artificial Unintelligence -- 11.6 Results -- 11.7 Lessons Learned -- 11.8 Conclusions -- Appendix 1 -- Appendix 2 -- Appendix 3 -- Appendix 4.

References -- 12 Quantum Computing Meets Artificial Intelligence: Innovations and Challenges -- 12.1 Introduction -- 12.1.1 Benefits of Quantum Computing for AI -- 12.2 Quantum Computing Motivations -- 12.2.1 What Does "Quantum" Mean? -- 12.2.2 The Wave-Particle Duality -- 12.2.3 Qubit Definition -- 12.2.4 The Schrödinger Equation -- 12.2.5 Superposition -- 12.2.6 Interference -- 12.2.7 Entanglement -- 12.2.8 Gate-Based Quantum Computing -- 12.3 Quantum Machine Learning -- 12.3.1 Variational Quantum Algorithms -- 12.3.2 Data Encoding -- 12.3.3 Quantum Neural Networks -- 12.3.4 Quantum Support Vector Machine -- 12.3.5 Variational Quantum Generator -- 12.4 Quantum Computing Limitations and Challenges -- 12.4.1 Scalability and Connectivity -- 12.4.2 Decoherence -- 12.4.3 Error Correction -- 12.4.4 Qubit Control -- 12.5 Quantum AI Software Engineering -- 12.5.1 Hybrid Quantum-Classical Frameworks -- 12.5.2 Friction-Less Development Environment -- 12.5.3 Quantum AI Software Life Cycle -- 12.6 A new Problem Solving Approach -- 12.6.1 Use Case 1: Automation and Transportation Sector -- 12.6.2 Use Case 2: Food for the Future World -- 12.6.3 Use Case 3: Cheaper Reliable Batteries -- 12.6.4 Use Case 4: Cleaner Air to Breathe -- 12.6.5 Use Case 5: AI-Driven Financial Solutions -- 12.7 Summary and Conclusion

-- References.
