

ai and software engineering

AI and Software Engineering: A Transformative Partnership

Author: Dr. Anya Sharma, PhD in Computer Science, Senior Research Scientist at Google AI, and author of "The Algorithmic Architect: Designing for AI-Powered Systems."

Keywords: AI and software engineering, AI in software development, software engineering challenges, AI opportunities, AI-powered tools, future of software engineering, AI ethics, AI safety, machine learning engineering, deep learning in software.

Abstract: This article explores the rapidly evolving relationship between AI and software engineering, examining both the exciting opportunities and the significant challenges this convergence presents. We will delve into how AI is transforming software development processes, from code generation to testing and deployment, while also considering the ethical and practical implications of integrating AI into the software engineering lifecycle.

1. Introduction: The AI Revolution in Software Engineering

The integration of Artificial Intelligence (AI) into software engineering is no longer a futuristic vision; it's a rapidly unfolding reality. AI is fundamentally altering how software is designed, developed, tested, and deployed. This symbiotic relationship between AI and software engineering presents a plethora of opportunities for increased efficiency, reduced costs, and the creation of more sophisticated and innovative software solutions. However, it also introduces new challenges, particularly concerning the reliability, security, and ethical implications of AI-powered tools. This article aims to provide a balanced perspective on this dynamic interplay, exploring both the transformative potential and the complexities inherent in the convergence of AI and software engineering.

1. AI-Powered Tools Transforming Software Development

Several AI-powered tools are already making significant inroads into various stages of the software development lifecycle (SDLC). These include:

AI-Assisted Code Generation: Tools using techniques like large language models (LLMs) can generate code snippets, entire functions, or even complete modules based on natural language descriptions. This significantly accelerates development, reduces human error, and allows developers to focus on higher-level design and problem-solving.

Automated Code Testing and Debugging: AI algorithms can analyze code for bugs, vulnerabilities, and potential performance issues, significantly improving the quality and reliability of software. Automated testing powered by AI can also adapt and learn from previous test runs, optimizing the testing process over time.

Intelligent Code Completion and Suggestions: AI-powered IDEs (Integrated Development Environments) provide intelligent code completion suggestions, reducing the time and effort required for writing code. These suggestions are often context-aware, improving both efficiency and code quality.

AI-Driven Project Management and Planning: AI can analyze project data, predict potential risks, and optimize resource allocation, leading to more efficient project management and improved delivery timelines.

Automated Deployment and Infrastructure Management: AI can automate the deployment process, scaling infrastructure resources dynamically based on demand and ensuring optimal performance. This reduces manual effort and improves the reliability of deployment pipelines.

1. Challenges in Integrating AI into Software Engineering

Despite the numerous benefits, integrating AI into software engineering presents several significant challenges:

Data Dependency: AI models require large amounts of high-quality data for training and effective operation. Acquiring, cleaning, and managing this data can be a significant challenge, particularly for specialized domains.

Explainability and Interpretability: Many AI models, especially deep learning models, are often "black boxes," making it difficult to understand their decision-making processes. This lack of transparency can be a major concern in safety-critical applications where understanding the reasoning behind an AI's actions is crucial.

Bias and Fairness: AI models can inherit biases present in the training data, leading to unfair or discriminatory outcomes. Ensuring fairness and mitigating bias in AI-powered software engineering tools is a critical ethical consideration.

Security and Robustness: AI models can be vulnerable to adversarial attacks, where malicious actors manipulate inputs to cause the model to behave unexpectedly. Ensuring the security and robustness of AI-powered software engineering tools is paramount.

Integration Complexity: Integrating AI tools into existing software development workflows can be complex and require significant effort, particularly for organizations with legacy systems.

1. The Future of AI and Software Engineering

The convergence of AI and software engineering is still in its early stages, but its transformative potential is undeniable. We can expect to see:

Increased Automation: More tasks in the SDLC will become automated, freeing up software engineers to focus on higher-level design, innovation, and problem-solving.

Enhanced Productivity: AI-powered tools will dramatically improve the productivity of software engineers, leading to faster development cycles and reduced costs.

Improved Software Quality: AI-driven testing and debugging will lead to higher-quality, more reliable, and more secure software.

New Software Paradigms: AI will enable the development of entirely new types of software, such as self-healing systems, adaptive applications, and intelligent user interfaces.

1. Ethical Considerations in AI and Software Engineering

The development and deployment of AI-powered software engineering tools necessitate careful consideration of ethical implications. This includes addressing issues of bias, fairness, transparency, accountability, and privacy. Robust ethical guidelines and regulations are crucial to ensure that AI is used responsibly in software engineering.

1. Conclusion: Embracing the Transformative Power of AI

The relationship between AI and software engineering is a powerful and transformative one. While challenges remain, the opportunities for increased efficiency, improved software quality, and the creation of innovative applications are immense. By carefully addressing the ethical and practical challenges, we can harness the power of AI to usher in a new era of software development, one characterized by greater productivity, enhanced reliability, and the creation of truly intelligent and user-centric systems. The future of software engineering is inextricably linked with the advancement of AI, and a collaborative approach focusing on responsible innovation is key to maximizing the benefits of this powerful partnership.

FAQs:

1. What are the most popular AI-powered tools used in software engineering today?
Popular tools include GitHub Copilot, Tabnine, DeepCode, and various AI-powered

testing frameworks.

1. How can I learn more about AI and software engineering? Online courses, workshops, and conferences focusing on AI, machine learning, and software engineering are excellent resources.

1. What are the potential job roles in the field of AI and software engineering? Roles include AI/ML engineers, data scientists, software engineers specializing in AI, and AI ethics specialists.

1. What are the biggest ethical concerns related to AI in software engineering? Bias in algorithms, lack of transparency, data privacy, and potential for misuse are major concerns.

1. How can I contribute to the ethical development of AI in software engineering? Advocating for responsible AI practices, participating in ethical discussions, and promoting transparency are important steps.

1. Will AI replace software engineers? AI will likely automate certain tasks, but it will also create new opportunities and augment the capabilities of human software engineers.

1. What are the key skills needed for a career in AI and software engineering? Strong programming skills, knowledge of machine learning and AI concepts, and an understanding of software engineering principles are essential.

1. How is AI improving the security of software systems? AI is used to detect vulnerabilities, predict attacks, and enhance security protocols.

1. What are the future trends in AI and software engineering? Expect more automation, increased focus on explainable AI, and the emergence of new programming

paradigms.

Related Articles:

1. "The Impact of AI on Software Testing": Explores how AI is revolutionizing software testing methodologies and improving test automation.
1. "AI-Powered Code Generation: A Deep Dive": A detailed analysis of the techniques and challenges associated with AI-assisted code generation.
1. "Ethical Considerations in AI-Driven Software Development": Focuses on the ethical implications of using AI in software development, emphasizing fairness, transparency, and accountability.
1. "The Future of Software Development in the Age of AI": Predicts future trends and challenges in the software development landscape, focusing on the role of AI.
1. "AI and DevOps: A Synergistic Partnership": Discusses how AI is transforming DevOps practices, improving automation and efficiency.
1. "Mitigating Bias in AI-Powered Software Engineering Tools": Explores methods and techniques for addressing bias in AI models used in software development.
1. "Explainable AI (XAI) in Software Engineering": Examines the importance of explainability in AI models used for software development and testing.
1. "Securing AI-Powered Software Systems": Focuses on the security challenges posed by AI and explores techniques for securing AI-driven software.

1. "Case Studies: Successful Applications of AI in Software Engineering": Presents real-world examples of how AI is being successfully used in various software engineering projects.

Publisher: IEEE Software, a leading publication in the field of software engineering, known for its rigorous peer-review process and high-quality technical articles.

Editor: Dr. David Miller, Professor of Computer Science at Stanford University, and expert in AI and software engineering.

ai and software engineering: *Artificial Intelligence Methods For Software Engineering* Meir Kalech, Rui Abreu, Mark Last, 2021-06-15 Software is an integral part of our lives today. Modern software systems are highly complex and often pose new challenges in different aspects of Software Engineering (SE). Artificial Intelligence (AI) is a growing field in computer science that has been proven effective in applying and developing AI techniques to address various SE challenges. This unique compendium covers applications of state-of-the-art AI techniques to the key areas of SE (design, development, debugging, testing, etc). All the materials presented are up-to-date. This reference text will benefit researchers, academics, professionals, and postgraduate students in AI, machine learning and software engineering. Related Link(s)

ai and software engineering: *Artificial Intelligence and Software Engineering* Derek Partridge, 2013-04-11 Managers, business owners, computer literate individuals, software developers, students, and researchers--all are looking for an understanding of artificial intelligence (AI) and what might be in the future. In this literate yet easy-to-read discussion, Derek Partridge explains what artificial intelligence can and cannot do, and what it holds for applications such as banking, financial services, and expert systems of all kinds. Topics include: the strengths and weaknesses of software development and engineering; machine learning and its promises and problems; expert systems and success stories; and practical software through artificial intelligence.

ai and software engineering: Readings in Artificial Intelligence and Software Engineering Charles Rich, Richard C. Waters, 2014-06-28 Readings in Artificial Intelligence and Software Engineering covers the main techniques and application of artificial intelligence and software engineering. The ultimate goal of artificial intelligence applied to software engineering is automatic programming. Automatic programming would allow a user to simply say what is wanted and have a program produced completely automatically. This book is organized into 11 parts encompassing 34 chapters that specifically tackle the topics of deductive synthesis, program transformations, program verification, and programming tutors. The opening parts provide an introduction to the key ideas to the deductive approach, namely the correspondence between theorems and specifications and between constructive proofs and programs. These parts also describes automatic theorem provers whose development has been designed for the programming domain. The subsequent parts present generalized program transformation systems, the problems involved in using natural language input, the features of very high level languages, and the advantages of the programming by example system. Other parts explore the intelligent assistant approach and the significance and relation of programming knowledge in other programming system. The concluding parts focus on the features of the domain knowledge system and the artificial intelligence programming. Software engineers and designers and computer programmers,

as well as researchers in the field of artificial intelligence will find this book invaluable.

ai and software engineering: *Software Engineering: Artificial Intelligence, Compliance, and Security* Brian D'Andrade, 2021-02-16 Information security is important in every aspect of daily life. This book examines four areas where risks are present: artificial intelligence (AI), the internet of things (IoT), government and malware. The authors channel their experience and research into an accessible body of knowledge for consideration by professionals. AI is introduced as a tool for healthcare, security and innovation. The advantages of using AI in new industries are highlighted in the context of recent developments in mechanical engineering, and a survey of AI software risks is presented focusing on well-publicized failures and US FDA regulatory guidelines. The risks associated with the billions of devices that form the IoT grow with the availability of such devices in consumer products, healthcare, energy infrastructure and transportation. The risks, software engineering risk mitigation methods and standards promoting a level of care for the manufacture of IoT devices are examined because of their importance for software developers. Strategic insights for software developers looking to do business with the US federal government are presented, considering threats to both public and private sectors as well as governmental priorities from recent executive and legislative branch actions. Finally, an analysis of malicious software that infects numerous computer systems each day and causes millions of dollars in damages every year is presented. Malicious software, or malware, is software designed with hostile intent, but the damage may be mitigated with static and dynamic analyses, which are processes for studying how malware operates and analyzing potential impacts.

ai and software engineering: *2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014)* S. K. Chen, Altair Engineering Inc., California, USA, Y. H. Chang, Chihlee Institute of Technology, Taiwan, 2014-02-06 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014) aims to provide a forum for accessing to the most up-to-date and authoritative knowledge from both Artificial Intelligence and Software Engineering. AISE2014 features unique mixed topics of AI Algorithms, Data Mining, Knowledge-based Systems, Software Process and so on. The goal of this conference is to bring researchers, engineers, and students to the areas of Artificial Intelligence and Software Engineering to share experiences and original research contributions on those topics. Researchers and practitioners are invited to submit their contributions to AISE2014.

ai and software engineering: Generative AI for Effective Software Development Anh Nguyen-Duc,

ai and software engineering: *Artificial Intelligence and Software Engineering* Derek Partridge, 1998

ai and software engineering: Code Convergence Instabooks Ai, 2024-03-02 Code Convergence: Navigating the New Era of AI in Software Engineering The rapid advancement of artificial intelligence in generating code poses both challenges and opportunities for software engineers. Code Convergence dives deep into the implications of this technological evolution, offering readers a comprehensive guide to staying relevant and harnessing AI's power in software development. This illuminating read is meticulously designed for everyone from budding engineers to seasoned software wizards, exploring the profound changes and strategies needed in the age of automation. Embark on a journey through thoughtfully crafted chapters that unravel the complexities of AI's role in software engineering. Discover practical strategies for leveraging AI tools, learn how to enhance your software design with AI's capabilities, and understand the evolving role of software engineers augmented by machine intelligence. With a blend of theoretical insights and real-world applications, Code Convergence empowers engineers to adapt and thrive in this transformative landscape. From comprehending how AI-generated code is changing the software development lifecycle, to exploring methods for integrating AI seamlessly into your projects, this book covers the spectrum of necessity and innovation. Uncover the secrets of staying ahead in your career by embracing and enhancing AI technologies, ensuring your expertise remains indispensable in the future of engineering. The realm of software engineering is at a pivotal turning point, and

Code Convergence serves as your guide. Prepare to explore, understand, and revolutionize your approach to software development with AI as your ally. The future beckons, and it's filled with endless possibilities for those ready to embrace the change.

Table of Contents

1. The AI Revolution in Software Development - The Rise of AI-Generated Code - Strategies for Working Alongside AI - Implications for the Software Development Lifecycle
2. Staying Ahead in the Age of Automation - Adapting to the New Normal - Leveraging AI for Career Growth - Maintaining Relevance Amid Rapid Change
3. Enhancing Software Design with AI - Integrating AI into Development Processes - The Impact of AI on Creativity - Optimizing Performance with AI Tools
4. The Evolving Role of Software Engineers - From Coding to Complex Problem-Solving - Augmenting Skills with AI - The Future of Engineering Work
5. Practical Strategies for Leveraging AI - AI Tools Every Engineer Should Know - Implementing AI in Your Projects - Case Studies and Success Stories
6. Understanding AI's Impact on Code Quality - Comparative Analysis of Human vs. AI Code - Improving Code Efficiency with AI - Future Trends in Code Generation
7. Ethical Considerations in AI Development - Evaluating Bias in AI-Generated Code - Privacy and Security Challenges - Responsibility and Accountability of Engineers
8. Collaboration and Innovation with AI - Building AI-Enhanced Software Teams - Interdisciplinary Approaches to Development - Fostering a Culture of Continuous Learning
9. The Global Landscape of AI and Software Engineering - International Trends in AI Adoption - Cultural Influences on AI Development - Global Opportunities for Engineers
10. Navigating Career Pathways in AI-Era Engineering - Skills for the Future - Building a Portfolio of AI-Enhanced Projects - Networking and Community Engagement
11. AI and the Future of Software Design - Evolution of Design Paradigms - AI as a Design Partner - Redefining Creativity in Engineering
12. Conclusion: Embracing the Shift - Summarizing the Journey - Preparing for What Comes Next - The Endless Possibilities of AI in Engineering

ai and software engineering: Handbook on Artificial Intelligence-Empowered Applied Software Engineering Maria Virvou, George A. Tsihrintzis, Nikolaos G. Bourbakis, Lakhmi C. Jain, 2022-10-17 Evolving technological advancements in big data, smartphone and mobile software applications, the Internet of Things and a vast range of application areas in all sorts of human activities and professions, lead current research toward the efficient incorporation of artificial intelligence enhancements into software and the empowerment of software with artificial intelligence. The book at hand, devoted to Smart Software Applications in Cyber-Physical Systems, constitutes the second volume of a two-volume Handbook on Artificial Intelligence-empowered Applied Software Engineering. Topics include very significant advances in Smart Software Applications in (i) Scientific Document Processing, (ii) Enterprise Modeling, (iii) Education, (iv) Health care and Medicine, and (v) Infrastructure Monitoring. Professors, researchers, scientists, engineers, and students in artificial intelligence, software engineering, and computer science-related disciplines are expected to benefit from it, along with interested readers from other disciplines.

ai and software engineering: Machine Learning Applications In Software Engineering Du Zhang, Jeffrey J P Tsai, 2005-02-21 Machine learning deals with the issue of how to build computer programs that improve their performance at some tasks through experience. Machine learning algorithms have proven to be of great practical value in a variety of application domains. Not surprisingly, the field of software engineering turns out to be a fertile ground where many software development and maintenance tasks could be formulated as learning problems and approached in terms of learning algorithms. This book deals with the subject of machine learning applications in software engineering. It provides an overview of machine learning, summarizes the state-of-the-practice in this niche area, gives a classification of the existing work, and offers some application guidelines. Also included in the book is a collection of previously published papers in this research area.

ai and software engineering: Deep Learning for Coders with fastai and PyTorch Jeremy Howard, Sylvain Gugger, 2020-06-29 Deep learning is often viewed as the exclusive domain of math PhDs and big tech companies. But as this hands-on guide demonstrates, programmers comfortable

with Python can achieve impressive results in deep learning with little math background, small amounts of data, and minimal code. How? With fastai, the first library to provide a consistent interface to the most frequently used deep learning applications. Authors Jeremy Howard and Sylvain Gugger, the creators of fastai, show you how to train a model on a wide range of tasks using fastai and PyTorch. You'll also dive progressively further into deep learning theory to gain a complete understanding of the algorithms behind the scenes. Train models in computer vision, natural language processing, tabular data, and collaborative filtering Learn the latest deep learning techniques that matter most in practice Improve accuracy, speed, and reliability by understanding how deep learning models work Discover how to turn your models into web applications Implement deep learning algorithms from scratch Consider the ethical implications of your work Gain insight from the foreword by PyTorch cofounder, Soumith Chintala

ai and software engineering: Human-Centered AI Ben Shneiderman, 2022 The remarkable progress in algorithms for machine and deep learning have opened the doors to new opportunities, and some dark possibilities. However, a bright future awaits those who build on their working methods by including HCAI strategies of design and testing. As many technology companies and thought leaders have argued, the goal is not to replace people, but to empower them by making design choices that give humans control over technology. In Human-Centered AI, Professor Ben Shneiderman offers an optimistic realist's guide to how artificial intelligence can be used to augment and enhance humans' lives. This project bridges the gap between ethical considerations and practical realities to offer a road map for successful, reliable systems. Digital cameras, communications services, and navigation apps are just the beginning. Shneiderman shows how future applications will support health and wellness, improve education, accelerate business, and connect people in reliable, safe, and trustworthy ways that respect human values, rights, justice, and dignity.

ai and software engineering: Engineering Artificial Intelligence Software Derek Partridge, 1992 This book explains the reality of exploiting the promise of AI in software systems. It presents the realities, the problems, the current state of the art, and future directions. It includes an examination of the problems that engineering AI software involves, and a consideration of the alternative routes to solution of these problems.

ai and software engineering: Artificial Intelligence Derek Partridge, 1986

ai and software engineering: Software Engineering at Google Titus Winters, Tom Manshreck, Hyrum Wright, 2020-02-28 Today, software engineers need to know not only how to program effectively but also how to develop proper engineering practices to make their codebase sustainable and healthy. This book emphasizes this difference between programming and software engineering. How can software engineers manage a living codebase that evolves and responds to changing requirements and demands over the length of its life? Based on their experience at Google, software engineers Titus Winters and Hyrum Wright, along with technical writer Tom Manshreck, present a candid and insightful look at how some of the world's leading practitioners construct and maintain software. This book covers Google's unique engineering culture, processes, and tools and how these aspects contribute to the effectiveness of an engineering organization. You'll explore three fundamental principles that software organizations should keep in mind when designing, architecting, writing, and maintaining code: How time affects the sustainability of software and how to make your code resilient over time How scale affects the viability of software practices within an engineering organization What trade-offs a typical engineer needs to make when evaluating design and development decisions

ai and software engineering: Skills of a Successful Software Engineer Fernando Doglio, 2022-08-16 Skills to grow from a solo coder into a productive member of a software development team, with seasoned advice on everything from refactoring to acing an interview. In Skills of a Successful Software Engineer you will learn: The skills you need to succeed on a software development team Best practices for writing maintainable code Testing and commenting code for others to read and use Refactoring code you didn't write What to expect from a technical interview

process How to be a tech leader Getting around gatekeeping in the tech community Skills of a Successful Software Engineer is a best practices guide for succeeding on a software development team. The book reveals how to optimize both your code and your career, from achieving a good work-life balance to writing the kind of bug-free code delivered by pros. You'll master essential skills that you might not have learned as a solo coder, including meaningful code commenting, unit testing, and using refactoring to speed up feature delivery. Timeless advice on acing interviews and setting yourself up for leadership will help you throughout your career. Crack open this one-of-a-kind guide, and you'll soon be working in the professional manner that software managers expect. About the technology Success as a software engineer requires technical knowledge, flexibility, and a lot of persistence. Knowing how to work effectively with other developers can be the difference between a fulfilling career and getting stuck in a life-sucking rut. This brilliant book guides you through the essential skills you need to survive and thrive on a software engineering team. About the book Skills of a Successful Software Engineer presents techniques for working on software projects collaboratively. In it, you'll build technical skills, such as writing simple code, effective testing, and refactoring, that are essential to creating software on a team. You'll also explore soft skills like how to keep your knowledge up to date, interacting with your team leader, and even how to get a job you'll love. What's inside Best practices for writing and documenting maintainable code Testing and refactoring code you didn't write What to expect in a technical interview How to thrive on a development team About the reader For working and aspiring software engineers. About the author Fernando Doglio has twenty years of experience in the software industry, where he has worked on everything from web development to big data. Table of Contents 1 Becoming a successful software engineer 2 Writing code everyone can read 3 Unit testing: delivering code that works 4 Refactoring existing code (or Refactoring doesn't mean rewriting code) 5 Tackling the personal side of coding 6 Interviewing for your place on the team 7 Working as part of a team 8 Understanding team leadership

ai and software engineering: *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects* Meziane, Farid, Vadera, Sunil, 2009-07-31 This book provides an overview of useful techniques in artificial intelligence for future software development along with critical assessment for further advancement--Provided by publisher.

ai and software engineering: *Exploring Artificial Intelligence* Howard E. Shrobe, 2014-05-12 Exploring Artificial Intelligence: Survey Talks from the National Conference on Artificial Intelligence provides information pertinent to the distinct subareas of artificial intelligence research. This book discusses developments in machine learning techniques. Organized into six parts encompassing 16 chapters, this book begins with an overview of intelligent tutoring systems, which describes how to guide a student to learn new concepts. This text then links closely with one of the concerns of intelligent tutoring systems, namely how to interact through the utilization of natural language. Other chapters consider the various aspects of natural language understanding and survey the huge body of work that tries to characterize heuristic search programs. This book discusses as well how computer programs can create plans to satisfy goals. The final chapter deals with computational facilities that support. This book is a valuable resource for cognitive scientists, psychologists, domain experts, computer scientists, instructional designers, expert teachers, and research workers.

ai and software engineering: *Experimentation in Software Engineering* Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, Anders Wesslén, 2012-06-16 Like other sciences and engineering disciplines, software engineering requires a cycle of model building, experimentation, and learning. Experiments are valuable tools for all software engineers who are involved in evaluating and choosing between different methods, techniques, languages and tools. The purpose of Experimentation in Software Engineering is to introduce students, teachers, researchers, and practitioners to empirical studies in software engineering, using controlled experiments. The introduction to experimentation is provided through a process perspective, and the focus is on the steps that we have to go through to perform an experiment. The book is divided into

three parts. The first part provides a background of theories and methods used in experimentation. Part II then devotes one chapter to each of the five experiment steps: scoping, planning, execution, analysis, and result presentation. Part III completes the presentation with two examples. Assignments and statistical material are provided in appendixes. Overall the book provides indispensable information regarding empirical studies in particular for experiments, but also for case studies, systematic literature reviews, and surveys. It is a revision of the authors' book, which was published in 2000. In addition, substantial new material, e.g. concerning systematic literature reviews and case study research, is introduced. The book is self-contained and it is suitable as a course book in undergraduate or graduate studies where the need for empirical studies in software engineering is stressed. Exercises and assignments are included to combine the more theoretical material with practical aspects. Researchers will also benefit from the book, learning more about how to conduct empirical studies, and likewise practitioners may use it as a "cookbook" when evaluating new methods or techniques before implementing them in their organization.

ai and software engineering: *Handbook on Artificial Intelligence-Empowered Applied Software Engineering* Maria Virvou, George A. Tsihrintzis, Nikolaos G. Bourbakis, Lakhmi C. Jain, 2022-09-03 This book provides a structured overview of artificial intelligence-empowered applied software engineering. Evolving technological advancements in big data, smartphone and mobile software applications, the Internet of Things and a vast range of application areas in all sorts of human activities and professions lead current research towards the efficient incorporation of artificial intelligence enhancements into software and the empowerment of software with artificial intelligence. This book at hand, devoted to Novel Methodologies to Engineering Smart Software Systems Novel Methodologies to Engineering Smart Software Systems, constitutes the first volume of a two-volume Handbook on Artificial Intelligence-empowered Applied Software Engineering. Topics include very significant advances in (i) Artificial Intelligence-Assisted Software Development and (ii) Software Engineering Tools to develop Artificial Intelligence Applications, as well as a detailed Survey of Recent Relevant Literature. Professors, researchers, scientists, engineers and students in artificial intelligence, software engineering and computer science-related disciplines are expected to benefit from it, along with interested readers from other disciplines.

ai and software engineering: *Why AI Will Not Eliminate Software Engineering Jobs* Mohammad Zaripour, 2024-08-18 Artificial intelligence (AI) has quickly moved from science fiction to reality, transforming industries like healthcare, finance, manufacturing, and entertainment. With its ability to analyze vast amounts of data, recognize patterns, and make decisions, AI is boosting efficiency and productivity across the board. However, the rise of AI has also sparked fear and uncertainty, particularly in software engineering. Many worry that AI will take over human jobs, rendering software engineers obsolete. This book, *Why AI Will Not Eliminate Software Engineering Jobs*, aims to address these fears and provide a balanced perspective. While AI will undoubtedly change the landscape of software engineering, it will not make human engineers redundant. Instead, AI will enhance human capabilities, allowing engineers to focus on more complex and creative aspects of their work. Software engineering is much more than just writing code; it involves creativity, problem-solving, and understanding user needs-areas where human engineers excel. AI can handle repetitive and data-intensive tasks, but it lacks the nuanced thinking and emotional intelligence that humans bring to the table. Throughout this book, we explore the strengths and limitations of AI, the critical role of human software engineers, and the symbiotic relationship between AI and humans. We present real-world examples of effective collaboration between AI and engineers and discuss the evolving role of software engineers in an AI-augmented world. By the end of this book, readers will have a clear understanding of why AI will not eliminate software engineering jobs and how they can leverage AI to enhance their careers. This book is here to reassure readers that by embracing AI as a tool and continuously developing their skills, they can remain relevant and competitive in the job market. The human touch is invaluable, and the unique abilities of software engineers will always be in demand. As we embark on this journey, readers are encouraged to see AI not as a threat but as an ally. Together, we will explore the exciting

possibilities that lie ahead and how to thrive in a world where AI and human ingenuity work hand in hand.

ai and software engineering: Building Intelligent Systems Geoff Hulten, 2018-03-06 Produce a fully functioning Intelligent System that leverages machine learning and data from user interactions to improve over time and achieve success. This book teaches you how to build an Intelligent System from end to end and leverage machine learning in practice. You will understand how to apply your existing skills in software engineering, data science, machine learning, management, and program management to produce working systems. Building Intelligent Systems is based on more than a decade of experience building Internet-scale Intelligent Systems that have hundreds of millions of user interactions per day in some of the largest and most important software systems in the world. What You'll Learn Understand the concept of an Intelligent System: What it is good for, when you need one, and how to set it up for success Design an intelligent user experience: Produce data to help make the Intelligent System better over time Implement an Intelligent System: Execute, manage, and measure Intelligent Systems in practice Create intelligence: Use different approaches, including machine learning Orchestrate an Intelligent System: Bring the parts together throughout its life cycle and achieve the impact you want Who This Book Is For Software engineers, machine learning practitioners, and technical managers who want to build effective intelligent systems

ai and software engineering: Advancing Software Engineering Through AI, Federated Learning, and Large Language Models Sharma, Avinash Kumar, Chanderwal, Nitin, Prajapati, Amarjeet, Singh, Pancham, Kansal, Mrignainy, 2024-05-02 The rapid evolution of software engineering demands innovative approaches to meet the growing complexity and scale of modern software systems. Traditional methods often need help to keep pace with the demands for efficiency, reliability, and scalability. Manual development, testing, and maintenance processes are time-consuming and error-prone, leading to delays and increased costs. Additionally, integrating new technologies, such as AI, ML, Federated Learning, and Large Language Models (LLM), presents unique challenges in terms of implementation and ethical considerations. Advancing Software Engineering Through AI, Federated Learning, and Large Language Models provides a compelling solution by comprehensively exploring how AI, ML, Federated Learning, and LLM intersect with software engineering. By presenting real-world case studies, practical examples, and implementation guidelines, the book ensures that readers can readily apply these concepts in their software engineering projects. Researchers, academicians, practitioners, industrialists, and students will benefit from the interdisciplinary insights provided by experts in AI, ML, software engineering, and ethics.

ai and software engineering: Artificial intelligence & software engineering Derek Partridge, 1988

ai and software engineering: Modern Software Engineering David Farley, 2021-11-16 Improve Your Creativity, Effectiveness, and Ultimately, Your Code In Modern Software Engineering, continuous delivery pioneer David Farley helps software professionals think about their work more effectively, manage it more successfully, and genuinely improve the quality of their applications, their lives, and the lives of their colleagues. Writing for programmers, managers, and technical leads at all levels of experience, Farley illuminates durable principles at the heart of effective software development. He distills the discipline into two core exercises: learning and exploration and managing complexity. For each, he defines principles that can help you improve everything from your mindset to the quality of your code, and describes approaches proven to promote success. Farley's ideas and techniques cohere into a unified, scientific, and foundational approach to solving practical software development problems within realistic economic constraints. This general, durable, and pervasive approach to software engineering can help you solve problems you haven't encountered yet, using today's technologies and tomorrow's. It offers you deeper insight into what you do every day, helping you create better software, faster, with more pleasure and personal fulfillment. Clarify what you're trying to accomplish Choose your tools based on sensible criteria Organize work and systems to facilitate continuing incremental progress Evaluate your progress

toward thriving systems, not just more legacy code Gain more value from experimentation and empiricism Stay in control as systems grow more complex Achieve rigor without too much rigidity Learn from history and experience Distinguish good new software development ideas from bad ones Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

ai and software engineering: Human Factors in Global Software Engineering Rehman, Mobashar, Amin, Aamir, Gilal, Abdul Rehman, Hashmani, Manzoor Ahmed, 2019-06-28 More software engineers are likely to work in a globally distributed environment, which brings benefits that include quick and better software development, less manpower retention, scalability, and less software development cost and sharing of knowledge from the global pool of employees. However, these work environments also introduce a physical separation between team members and project leaders, which can create problems in communication and ultimately lead to the failure of the project. Human Factors in Global Software Engineering is a collection of innovative research focusing on the challenges, issues, and importance of human factors in global software engineering organizations in order to help these organizations better manage their manpower and provide an appropriate culture and technology in order to make their software development projects successful. While highlighting topics including agile software, knowledge management, and human-computer interaction, this book is ideally designed for project managers, administrators, business professionals, researchers, practitioners, students, and academicians.

ai and software engineering: The Essence of Artificial Intelligence Alison Cawsey, 1998 A concise, practical introduction to artificial intelligence, this title starts with the fundamentals of knowledge representation, inference, expert systems, natural language processing, machine learning, neural networks, agents, robots, and much more. Examples and algorithms are presented throughout, and the book includes a complete glossary.

ai and software engineering: Artificial Intelligence and Software Engineering Derek Partridge, 2013-04-11 Managers, business owners, computer literate individuals, software developers, students, and researchers--all are looking for an understanding of artificial intelligence (AI) and what might be in the future. In this literate yet easy-to-read discussion, Derek Partridge explains what artificial intelligence can and cannot do, and what it holds for applications such as banking, financial services, and expert systems of all kinds. Topics include: the strengths and weaknesses of software development and engineering; machine learning and its promises and problems; expert systems and success stories; and practical software through artificial intelligence.

ai and software engineering: Data Democracy Feras A. Batarseh, Ruixin Yang, 2020-01-28 This book provides a manifesto to data democracy. After reading the chapters of this book, you are informed and suitably warned! You are already part of the data republic, and you (and all of us) need to ensure that our data fall in the right hands. Everything you click, buy, swipe, try, sell, drive, or fly is a data point. But who owns the data? At this point, not you! You do not even have access to most of it. The next best empire of our planet is one who owns and controls the world's best dataset. If you consume or create data, if you are a citizen of the data republic (willingly or grudgingly), and if you are interested in making a decision or finding the truth through data-driven analysis, this book is for you. A group of experts, academics, data science researchers, and industry practitioners gathered to write this manifesto about data democracy. - The future of the data republic, life within a data democracy, and our digital freedoms. - An in-depth analysis of open science, open data, open source software, and their future challenges. - A comprehensive review of data democracy's implications within domains such as: healthcare, space exploration, earth sciences, business, and psychology. - The democratization of Artificial Intelligence (AI), and data issues such as: bias, imbalance, context, and knowledge extraction. - A systematic review of AI methods applied to software engineering problems.

ai and software engineering: Systems Engineering and Artificial Intelligence William F. Lawless, Ranjeev Mittu, Donald A. Sofge, Thomas Shortell, Thomas A. McDermott, 2021-11-02 This book provides a broad overview of the benefits from a Systems Engineering design philosophy in

architecting complex systems composed of artificial intelligence (AI), machine learning (ML) and humans situated in chaotic environments. The major topics include emergence, verification and validation of systems using AI/ML and human systems integration to develop robust and effective human-machine teams—where the machines may have varying degrees of autonomy due to the sophistication of their embedded AI/ML. The chapters not only describe what has been learned, but also raise questions that must be answered to further advance the general Science of Autonomy. The science of how humans and machines operate as a team requires insights from, among others, disciplines such as the social sciences, national and international jurisprudence, ethics and policy, and sociology and psychology. The social sciences inform how context is constructed, how trust is affected when humans and machines depend upon each other and how human-machine teams need a shared language of explanation. National and international jurisprudence determine legal responsibilities of non-trivial human-machine failures, ethical standards shape global policy, and sociology provides a basis for understanding team norms across cultures. Insights from psychology may help us to understand the negative impact on humans if AI/ML based machines begin to outperform their human teammates and consequently diminish their value or importance. This book invites professionals and the curious alike to witness a new frontier open as the Science of Autonomy emerges.

ai and software engineering: How Computers Really Work Matthew Justice, 2020-12-29 An approachable, hands-on guide to understanding how computers work, from low-level circuits to high-level code. How Computers Really Work is a hands-on guide to the computing ecosystem: everything from circuits to memory and clock signals, machine code, programming languages, operating systems, and the internet. But you won't just read about these concepts, you'll test your knowledge with exercises, and practice what you learn with 41 optional hands-on projects. Build digital circuits, craft a guessing game, convert decimal numbers to binary, examine virtual memory usage, run your own web server, and more. Explore concepts like how to: Think like a software engineer as you use data to describe a real world concept Use Ohm's and Kirchhoff's laws to analyze an electrical circuit Think like a computer as you practice binary addition and execute a program in your mind, step-by-step The book's projects will have you translate your learning into action, as you: Learn how to use a multimeter to measure resistance, current, and voltage Build a half adder to see how logical operations in hardware can be combined to perform useful functions Write a program in assembly language, then examine the resulting machine code Learn to use a debugger, disassemble code, and hack a program to change its behavior without changing the source code Use a port scanner to see which internet ports your computer has open Run your own server and get a solid crash course on how the web works And since a picture is worth a thousand bytes, chapters are filled with detailed diagrams and illustrations to help clarify technical complexities. Requirements: The projects require a variety of hardware - electronics projects need a breadboard, power supply, and various circuit components; software projects are performed on a Raspberry Pi. Appendix B contains a complete list. Even if you skip the projects, the book's major concepts are clearly presented in the main text.

ai and software engineering: New Perspectives in Software Engineering Jezreel Mejía,
ai and software engineering: Model-Driven Software Engineering in Practice Marco Brambilla, Jordi Cabot, Manuel Wimmer, 2017-03-30 This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of

MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios, and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The second edition of the book features: a set of completely new topics, including: full example of the creation of a new modeling language (IFML), discussion of modeling issues and approaches in specific domains, like business process modeling, user interaction modeling, and enterprise architecture complete revision of examples, figures, and text, for improving readability, understandability, and coherence better formulation of definitions, dependencies between concepts and ideas addition of a complete index of book content In addition to the contents of the book, more resources are provided on the book's website <http://www.mdse-book.com>, including the examples presented in the book.

ai and software engineering: Artificial Intelligence in Structural Engineering Ian Smith, 1998-07-15 This book presents the state of the art of artificial intelligence techniques applied to structural engineering. The 28 revised full papers by leading scientists were solicited for presentation at a meeting held in Ascona, Switzerland, in July 1998. The recent advances in information technology, in particular decreasing hardware cost, Internet communication, faster computation, increased bandwidth, etc., allow for the application of new AI techniques to structural engineering. The papers presented deal with new aspects of information technology support for the design, analysis, monitoring, control and diagnosis of various structural engineering systems.

ai and software engineering: Applications of Artificial Intelligence in Process Systems Engineering Jingzheng Ren, Weifeng Shen, Yi Man, Lichun Dong, 2021-06-05 Applications of Artificial Intelligence in Process Systems Engineering offers a broad perspective on the issues related to artificial intelligence technologies and their applications in chemical and process engineering. The book comprehensively introduces the methodology and applications of AI technologies in process systems engineering, making it an indispensable reference for researchers and students. As chemical processes and systems are usually non-linear and complex, thus making it challenging to apply AI methods and technologies, this book is an ideal resource on emerging areas such as cloud computing, big data, the industrial Internet of Things and deep learning. With process systems engineering's potential to become one of the driving forces for the development of AI technologies, this book covers all the right bases. - Explains the concept of machine learning, deep learning and state-of-the-art intelligent algorithms - Discusses AI-based applications in process modeling and simulation, process integration and optimization, process control, and fault detection and diagnosis - Gives direction to future development trends of AI technologies in chemical and process engineering

ai and software engineering: Artificial Intelligence for Autonomous Networks Mazin Gilbert, 2018-09-25 Artificial Intelligence for Autonomous Networks introduces the autonomous network by juxtaposing two unique technologies and communities: Networking and AI. The book reviews the technologies behind AI and software-defined network/network function virtualization, highlighting the exciting opportunities to integrate those two worlds. Outlining the new frontiers for autonomous networks, this book highlights their impact and benefits to consumers and enterprise customers. It also explores the potential of the autonomous network for transforming network operation, cyber security, enterprise services, 5G and IoT, infrastructure monitoring and traffic optimization, and finally, customer experience and care. With contributions from leading experts, this book will provide an invaluable resource for network engineers, software engineers, artificial intelligence, and machine learning researchers.

ai and software engineering: Artificial Unintelligence Meredith Broussard, 2019-01-29 A guide to understanding the inner workings and outer limits of technology and why we should never assume that computers always get it right. In Artificial Unintelligence, Meredith Broussard argues

that our collective enthusiasm for applying computer technology to every aspect of life has resulted in a tremendous amount of poorly designed systems. We are so eager to do everything digitally—hiring, driving, paying bills, even choosing romantic partners—that we have stopped demanding that our technology actually work. Broussard, a software developer and journalist, reminds us that there are fundamental limits to what we can (and should) do with technology. With this book, she offers a guide to understanding the inner workings and outer limits of technology—and issues a warning that we should never assume that computers always get things right. Making a case against technochauvinism—the belief that technology is always the solution—Broussard argues that it's just not true that social problems would inevitably retreat before a digitally enabled Utopia. To prove her point, she undertakes a series of adventures in computer programming. She goes for an alarming ride in a driverless car, concluding “the cyborg future is not coming any time soon”; uses artificial intelligence to investigate why students can't pass standardized tests; deploys machine learning to predict which passengers survived the Titanic disaster; and attempts to repair the U.S. campaign finance system by building AI software. If we understand the limits of what we can do with technology, Broussard tells us, we can make better choices about what we should do with it to make the world better for everyone.

ai and software engineering: Design Thinking in Software and AI Projects Robert Stackowiak, Tracey Kelly, 2020-12-05 Learn the fundamentals of Design Thinking and how to apply Design Thinking techniques in defining software development and AI solutions. Design Thinking is an approach to innovation which identifies problems and generates solution ideas that can be rapidly proven through prototyping. This book provides a brief history of Design Thinking and an overview of the process. It then drills down into more detail regarding methods and tools used in a Design Thinking workshops leading to useful prototypes. Guidance is provided on: Preparing for a Design Thinking Workshop Uncovering potential business problems that might be solved Prioritizing potential solutions Identifying and characterizing stakeholders Choosing the right prototypes for development Limiting scope and best practices in prototype building The book concludes with a discussion of best practices in operationalizing successful prototypes, and describes change management techniques critical for successful adoption. You can use the knowledge gained from reading this book to incorporate Design Thinking techniques in your software development and AI projects, and assure timely and successful delivery of solutions. What You Will Learn Gain foundational knowledge of what Design Thinking is and when to apply the technique Discover preparation and facilitation techniques used in workshops Know how ideas are generated and then validated through prototyping Understand implementation best practices, including change management considerations Who This Book Is For Business decision makers and project stakeholders as well as IT project owners who seek a method leading to fast development of successful software and AI prototypes demonstrating real business value. Also for data scientists, developers, and systems integrators who are interested in facilitating or utilizing Design Thinking workshops to drive momentum behind potential software development and AI projects.

ai and software engineering: Guide to the Software Engineering Body of Knowledge (Swebok(r)) IEEE Computer Society, 2014 In the Guide to the Software Engineering Body of Knowledge (SWEBOK(R) Guide), the IEEE Computer Society establishes a baseline for the body of knowledge for the field of software engineering, and the work supports the Society's responsibility to promote the advancement of both theory and practice in this field. It should be noted that the Guide does not purport to define the body of knowledge but rather to serve as a compendium and guide to the knowledge that has been developing and evolving over the past four decades. Now in Version 3.0, the Guide's 15 knowledge areas summarize generally accepted topics and list references for detailed information. The editors for Version 3.0 of the SWEBOK(R) Guide are Pierre Bourque (Ecole de technologie superieure (ETS), Universite du Quebec) and Richard E. (Dick) Fairley (Software and Systems Engineering Associates (S2EA)).

ai and software engineering: Continuous Delivery Jez Humble, David Farley, 2010-07-27 Winner of the 2011 Jolt Excellence Award! Getting software released to users is often a painful,

risky, and time-consuming process. This groundbreaking new book sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours— sometimes even minutes—no matter what the size of a project or the complexity of its code base. Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the “deployment pipeline,” an automated process for managing all changes, from check-in to release. Finally, they discuss the “ecosystem” needed to support continuous delivery, from infrastructure, data and configuration management to governance. The authors introduce state-of-the-art techniques, including automated infrastructure management and data migration, and the use of virtualization. For each, they review key issues, identify best practices, and demonstrate how to mitigate risks. Coverage includes • Automating all facets of building, integrating, testing, and deploying software • Implementing deployment pipelines at team and organizational levels • Improving collaboration between developers, testers, and operations • Developing features incrementally on large and distributed teams • Implementing an effective configuration management strategy • Automating acceptance testing, from analysis to implementation • Testing capacity and other non-functional requirements • Implementing continuous deployment and zero-downtime releases • Managing infrastructure, data, components and dependencies • Navigating risk management, compliance, and auditing Whether you’re a developer, systems administrator, tester, or manager, this book will help your organization move from idea to release faster than ever—so you can deliver value to your business rapidly and reliably.

Additional Resources

AI for Software Engineering

AI for Software Engineering (AI4SE): A Blessing or a Curse? AI4SE has become an umbrella term to refer to research that uses AI approaches to tackle software engineering challenges. AI ...

Software Engineering for Machine Learning: A Case Study

We found that various Microsoft teams have united this workflow into preexisting, well-evolved, Agile-like software engineering processes, providing insights about several essential ...

The Role of Artificial Intelligence in Software Engineeri

The areas in which AI techniques have proved to be useful in software engineering research and practice can be characterised as ‘Probabilistic Software Engineering’, ‘Classification, Learning ...

Integrating Artificial Intelligence in Software Engineering...

Through a comprehensive analysis, this paper aims to provide a deeper understanding of the potential and challenges of AI in software engineering, offering insights into future research ...

Synergies Between Artificial Intelligence and Software ...

Jul 12, 2021 · In this chapter, we present a study of current AI trends in software engineering research, focused on the techniques more applicable to distinct aspects of the software process.

Software Engineering for AI-Based Systems: A Survey - arXiv.org

AI-based systems are software systems which include AI components. These systems learn by analyzing their environment and taking actions, aiming at having an intelligent behaviour.

Artificial Intelligence In Software Engineering: Integration And ...

This paper explores the integration of artificial intelligence (AI) into software engineering. It examines how AI can be effectively incorporated throughout the software development ...

Software Engineering Using Artificial Intelligence Techniques: ...

This paper relates AI techniques to software engineering processes specified by the IEEE 12207 standard of software engineering. The paper is driven by the activities and tasks specified in ...

Crafting the AI Software Engineer of the Future

Artificial intelligence (AI) technologies, particularly large language models, are transforming software engineering by automating coding tasks. However, software engineering extends ...

Turbocharging software with Gen AI - Capgemini

Build a repository of platforms and tools for a seamless and augmented software engineering experience. Privately and safely contextualize generative AI assistants with organization's own ...

Exploring the synergy between generative AI and software ...

In this paper, by outlining the application of generative AI in the indicated domains, we will show how generative AI optimizes, secures, and improves software engineering. When hybrid ...

Ways of Applying Artificial Intelligence in Software Engineering

The purpose with the Artificial Intelligence in Software Engineering Application Levels (AI-SEAL) taxonomy is to support researchers and practitioners to communicate, understand and discuss ...

AI Engineering: 11 Foundational Practices

AI system is a software-intensive system. The established principles of designing and deploying quality software systems that meet their mission goal on time apply to engineering AI systems. ...

What is Really Different in Engineering AI-Enabled Systems?

AI Engineering is a field of research and practice that combines the principles of systems engineering, software engineering, computer science, and human-centered design to create AI ...

Teaching Software Engineering for AI-Enabled Systems

Our course adopts a software-engineering perspective on building AI-enabled systems, focusing on what a software engineer can do to turn a machine learning idea into a scalable and ...

Application of Artificial Intelligence in Software Engineering

Artificial Intelligence techniques, which aim to create software systems that exhibit some

form of human intelligence, have been employed to assist or automate the activities in software ...

The Future of Software Engineering in an AI-Driven World

In this paper, we present our vision of the future of software development in an AI-driven world and explore the key challenges that our research community should address to realize this vision.

Trustworthy and Synergistic Artificial Intelligence for Software ...

As illustrated in Fig. 1, this paper first describes the history and key challenges of AI for Software Engineering (AI4SE) in Sections I and II, respectively. It then describes a vision for AI4SE in ...

Intelligent Software Engineering: Synergy Between AI and

The latter, [intelligence software] engineering, focuses on addressing various software engineering tasks for intelligence software, e.g., AI software. In this paper, we discuss recent ...

Experimenting with Multi-Agent Software Development: ...

studies concerning generative AI and its application across software engineering. Section 2.2 examines works that have utilized LLMs for software development. 2.1 Generative AI in ...

Agentic AI Software Engineer: Programming with Trust

So how can AI software engineers alleviate the issue of trust in automatically generated code? There can be varying levels of trust that the AI software engineer can establish. Testing for ...

Application of Artificial Intelligence in Software Engineering

The disciplines of artificial intelligence and software engineering have developed separately. There is not much exchange of research results between them. AI research techniques make ...

Artificial Intelligence in Software Testing : Impact, Problems ...

Jan 17, 2022 · nificant growth in the software industry driven by the recent advances in AI. Artificial Intelligence is gradually changing the landscape of software engineering in general ...

Revolutionizing Software Engineering with Generative AI: ...

Revolutionizing Software Engineering with Generative AI: Transforming Development Processes, Automation, and Quality Assurance Ankush Kesar Abstract: AI-integrated generative AI is a ...

Large Language Models for Software Engineering: Survey ...

Requirements engineering, Search Based Software Engineering (SBSE), Software Analytics, Software Engineering Education, Software Processes, Software Maintenance and Evolution, ...

AI-Tutoring in Software Engineering Education - arXiv.org

AI-Tutoring in Software Engineering Education Figure 1: Sequence diagram of the usage workflow of the Artemis system extended by the AI-Tutor functionality.

Before the integration of the AI ...

Towards AI-Native Software Engineering (SE 3.0): A Vision ...

The rise of AI-assisted software engineering (SE 2.0), powered by Foundation Models (FMs) and FM-powered copilots, has shown promise in improving developer productivity. However, it has ...

arXiv:2310.13976v2 [cs.SE] 1 Nov 2023

RE in this new era of generative AI-driven software engineering. This chapter explores the potential of LLMs to transform the RE processes. We present a SWOT (Strengths, ...

Artificial Intelligence in Software Requirements Engineering: ...

to engineering. Particularly in computing, AI has been applied to various fields such as data mining, speech and language processing, robotics, image recognition, visual analytics and ...

Software Engineering for AI-Based Systems: A Survey

This paper aims to characterize SE practices for AI-based systems in the new wave of AI, i.e., Software Engineering for Artificial Intelligence (SE4AI). The motivation of this work is to ...

Benchmarking AI Models in Software Engineering: A Review, ...

Benchmarking AI Models in Software Engineering: A Review, Search Tool, and Enhancement Protocol Roham Koohestani, Philippe de Bekker, and Maliheh Izadi Abstract—Benchmarks ...

REPOG : ENHANCING AI SOFTWARE ENGINEER R LEVEL ...

ern AI software engineering tasks. Unlike traditional function-level or file-level coding tasks, AI software engineering requires not only basic coding proficiency but also advanced skills in ...

Generative AI and Empirical Software Engineering: A ...

new opportunities for empirical software engineering research. Index Terms—Software Engineering, Generative AI, Empirical Methods. I. INTRODUCTION The software ...

Corporate overview Engineering modern businesses

on IoT, AI, software engineering and cloud— the technologies that are changing the nature of business. Today, creating value by leveraging technology is very industry-specific, so we ...

Guidelines for secure AI system development - Internet Crime ...

> Scale AI > Software Engineering Institute at Carnegie Mellon University > Stanford Center for AI Safety > Stanford Program on Geopolitics, ... AI systems are subject to novel security ...

AI Engineering: 11 Foundational Practices - Carnegie Mellon ...

Software Engineering Principles Apply to AI Engineering Along with the following recommendations, remember that an AI system is a software-intensive system. The ...

A.AVA: Revolutionizing Software Development with AI

AI-driven Automation: A.AVA leverages AI and ML to streamline manual tasks, improve efficiency, and continuously suggest optimization opportunities. Transparent Dashboard:

A.AVA's ...

FUTURE OF ARTIFICIAL INTELLIGENCE IN AGILE SOFTWARE ...

AI can assist software development managers, software testers, and other team members by leveraging LLMs, GenAI models, and AI agents to perform routine tasks, risk analysis and ...

SVR ENGINEERING COLLEGE

SOFTWARE ENGINEERING FOR AI (20A30502a) SVREC, NDL Page 1 JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR B.Tech CSE (AI)- III-I Sem L T P C 3 0 0 3 ...

Synergies Between Artificial Intelligence and Software ...

Jul 12, 2021 · Artificial intelligence (AI) for software engineering (SE) is expected to change the way we think about software planning, development and maintenance. Daily, soft-ware ...

The Role of Generative AI in Software Development ...

tices with the integration of generative AI tools. CCS CONCEPTS • Software and its engineering → Programming teams. KEYWORDS software engineering, generative AI, LLMs, productivity ...

AI-Driven Innovations in Software Engineering: A Review of ...

AI is revolutionizing software engineering by introducing automation and intelligence into the development lifecycle. Embracing AI-driven tools and methodologies is essential for staying ...

B.TECH COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL ...

PROGRAM SPECIFIC OUTCOMES FOR CSE-AI After completing the B.Tech CSE-AI program, the students will, 1. ... ENGG 22AIE311 Software Engineering (Project-Based) 2 0 3 3 ENGG ...

Agents in Software Engineering: Survey, Landscape, and ...

Agents in Software Engineering: Survey, Landscape, and Vision Yanlin Wang 1, Wanjun Zhong , Yanxian Huang *, Ensheng Shi2, Min Yang3, Jiachi Chen1, Hui Li4, Yuchi Ma5, Qianxiang ...

Human AI Collaboration in Software Engineering: Lessons ...

Human-AI Collaboration in Software Engineering: Lessons Learned from a Hands-On Workshop Muhammad Hamza Software Engineering LUT University Lahti, Finland ...

SWE-Lancer: Can Frontier LLMs Earn \$1 Million from Real ...

To better measure the real-world software engineering capabilities and impact of AI models, we introduce SWE-Lancer. Our benchmark evaluates frontier language models on 1,488 ...

AGENTLESS Demystifying LLM-based Software Engineering ...

Software engineering tasks like feature addition, program repair, ... inspired by the Devin AI Software Engineer [7], there has been a significant body of work from both academia and ...

Teaching Software Engineering for AI-Enabled Systems

or Siegmund's Search-Based Software Engineering at the University of Weimar. Our new

course has the opposite focus: how software engineering techniques can be used to build better ...

Artificial Intelligence and Software Engineering: Status ...

Agent-Oriented Software Engineering Software Agents are typically small intelligent systems that cooperate to reach a common goal. These agents are a relatively new area where research ...

Turbocharging software with Gen AI - Capgemini

Source: Capgemini Research Institute, Generative AI in Software Engineering, Senior Executive Survey, April 2024, n = 412 organizations represented by n = 412 software leaders that have ...

An Empirical Study on Usage and Perceptions of LLMs in a ...

cases of AI-generated code in an academic software engineering project. The code analysis across project phases and categorization of AI-generated code shows how LLMs could be ...

Assessing Opportunities for LLMs in Software Engineering ...

Table 1: Sample Acquisition and Software Engineering Use Cases ACQUISITION USE CASES SOFTWARE ENGINEERING USE CASES A1. A new acquisition specialist uses an LLM to ...

An Empirical Study on Decision-Making Aspects in ...

Jan 28, 2025 · the software engineering life cycle for AI. While ethical issues in responsible software engineering for AI largely mirror those found in broader software engineering ...

Introduction - cs.wm.edu

Generative AI in Software Engineering The integration of generative AI presents various opportunities in software engineering tasks, such as code generation, configuration ...

Advanced Approaches to Achieve Adaptive Ethical and AI ...

ethical human centric AI software engineering. By focusing on the integration of real-time active user feedback mechanisms and machine learning techniques, we seek to provide systems ...

A Systematic Literature Review of Explainable AI for Software ...

Feb 14, 2023 · A Systematic Literature Review of Explainable AI for Software Engineering Ahmad Haji Mohammadkhania, Nitin Sai Bommib, Mariem Daboussic, Onkar Sabnisd, Chakkrit ...

The Role of Artificial Intelligence and Machine Learning in ...

for dynamic test case generation. In this approach, AI agents interact with the software environment, learning from feedback to continuously improve test case generation and ...

Innovations in AI and Their Impact on Software Engineering ...

A. AI in Software Engineering Revolutionizing Software Engineering with AI The integration of AI in software engineering is revolutionizing the field by automating and optimizing various ...

The Agentic AI Mindset A Practitioner's Guide to Architectures ...

In software engineering, agentic AI architectures and design patterns are rapidly being applied to automate complex workflows, optimize resource allocation, enhance code

quality, and much ...

Introduction - Vanderbilt University

Generative AI in Software Engineering The integration of generative AI presents various opportunities in software engineering tasks, such as code generation, configuration ...

Working Paper 25-021 Generative AI and the Nature of Work

generative AI code completion tool for software developers. Leveraging millions of panel observations on work activities over a two year period, we use a program eligibility threshold to ...

Artificial Intelligence and Machine Learning Applied in ... - Ansys

Artificial Intelligence (AI) is the new shooting star in science and every industry has high hopes in using the “new” technology for almost everything. In this article, we would like to discuss what ...

Agile Processes in Software Engineering LNBIP 524 and ...

International Workshop on Advances in Software Intensive Startups Exploring the Potential of Generative AI: Use Cases in Software Startups 3 Mario Simaremare, Triando, and Sergio ...

Two gaping gaps in post- AI software engineering

The vanishing entry level Good old days Rise of AI Job market for new grads Demand for senior engineers

An Empirical Study on the Impact of Gender Diversity on ...

need for rigorous development and maintenance practices in AI software engineering. While previous research has explored gender diversity in soft-ware engineering, OSS, and AI ...

Geothermal Energy in the Era of Artificial Intelligence

AI Technology from Space, at Work Here on Earth Beyond Limits, a California-based AI software engineering company, is commercializing cognitive computing and AI software technologies ...

Steve Jobs: Pioneering AI in Software Engineering

The related work for "STEVE JOBS: Pioneering AI in Software Engineering" encompasses a broad range of studies and projects that provide context, inspiration, and insights into the ...

Guidelines for secure AI system development - U.S.

Nov 27, 2023 · > Scale AI > Software Engineering Institute at Carnegie Mellon University > Stanford Center for AI Safety > Stanford Program on Geopolitics, ... AI systems are subject to ...

Generative Artificial Intelligence: The Future of Model Based ...

Oct 11, 2023 · • The goal of AI4SE is to improve the efficiency, effectiveness, and quality of software engineering processes, by leveraging the capabilities of AI and machine learning to ...

[Ai And Software Engineering](#)

Ai And Software Engineering

Related Articles

- [ai solving math problems](#)
- [ai that answer any question](#)
- [ai financial statement analysis](#)

[Back to Home](#)