

Beyond Vibe Coding

Building Production-Grade Software with AI Agents and
Specification-Driven Development

David W. Liao

Beyond Vibe Coding

Building Production-Grade Software with AI Agents and Specification-Driven Development

About This Book

Beyond Vibe Coding explores how artificial intelligence is reshaping the craft of software development—from the creative flow of **Vibe Coding**, to the structured autonomy of **Agentic Coding**, to the rigor of **AI-assisted Specification-Driven Development (SDD)**. This book bridges imagination and engineering, showing how human creativity and machine intelligence can co-design reliable, scalable, and ethical software systems.

Designed for educators, students, and professionals alike, it offers a clear roadmap for understanding and mastering next-generation software development methodologies. Through practical examples, workflows, and real-world tools such as GitHub Spec Kit, readers will learn why and how to balance innovation with discipline—transforming speed into precision, and experimentation into sustainable software development practice.

Beyond Vibe Coding: Building Production-Grade Software with AI Agents and Specification-Driven Development

© 2025 David W. Liao

All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without prior written permission from the publisher, except in the case of brief quotations used in reviews or scholarly works.

ISBN: 979-8-271-00432-2 (paperback)

First Edition

Printed in the United States of America

Cover Design: AI-assisted, curated by the author

This work is intended for educational and informational use. The author and publisher make no representations or warranties regarding the completeness or accuracy of the content and disclaim any liability arising from its use.

All trademarks and product names mentioned are the property of their respective owners.

10 9 8 7 6 5 4 3 2 1

Table of Contents

Preface: From Speed to Structure — The New Era of AI-Powered Software Development.....	xi
A Note to the Instructor	xiii
Introduction: The Vibe - From Creativity to Execution	1
What is Vibe Coding?	1
The Evolution of Software Development.....	1
From Vibe to Specification: AI-Assisted SDD.....	2
Governance and the Developer's New Role	3
The Journey Ahead	3
Chapter 1: Generative AI and Large Language Models: The Engine of Vibe Coding	4
1.1 Demystifying Generative AI, Transformers, and LLMs	4
1.1.1 What is Generative AI?	5
1.1.2 How Transformers Work.....	5
1.1.3 Overview of LLMs	6
1.2 How LLMs Generate Code	7
1.2.1 Pattern Recognition and Prediction vs. Formal Specifications	7
1.2.2 Context-Aware Generation	7
1.2.3 Retrieval-Augmented Generation (RAG)	7
1.2.4 Multi-Agent Pipelines and Tool-Calling	7
1.3 Introduction to Prompt Engineering	8
1.3.1 Crafting Effective Prompts	8
1.3.2 The C.L.E.A.R. Framework.....	8
1.3.3 Chain-of-Thought Prompting	8
1.3.4 Iterative and Dialogic Prompting.....	8
1.4 Advanced Prompt Engineering — The Art of AI Communication	9
1.4.1 Beyond C.L.E.A.R.: Information-Dense Keywords & Context Priming	9
1.4.2 Managing Context Windows: /clear and Dynamic Context	10
1.4.3 Few-Shot Examples and Iterative Prompting	12
1.4.4 Prompt Versioning and Shared Libraries	14
1.5 Interpreting and Evaluating AI-Generated Outputs	16
1.5.1 Understanding AI "Hallucinations"	16
1.5.2 The Necessity of Human Review, Inspection, and Revision	16

1.5.3 Challenges in Comprehending and Debugging	16
1.6 Context Engineering: Beyond Prompts	16
1.6.1 Why Context Engineering?.....	16
1.6.2 Key Context Engineering Techniques	17
1.6.3 Example	18
1.7 Multimodal Capabilities of LLMs	20
1.8 Summary and Takeaways	20
Review Questions.....	23
Multiple-Choice Questions	23
Short-Answer Conceptual Questions	25
Applied Project-Scenario Questions	26
Discussion Questions	27
Practical Lab Tasks	28
Chapter 2: The End-to-End Vibe Coding Workflow: From Idea to Production	30
2.1 Setting the Stage for Vibe Coding	30
2.1.1 From Idea to Concrete Project: PRD, DoD and More	30
2.1.2 Context Priming: Providing Clear Goals, Constraints, and Artifacts	31
2.1.3 Brainstorming and Ideation with AI Tools	32
2.2 UI/UX Prototyping and Vibe Design	32
2.2.1 Dedicated AI Design Tools and Vibe Design.....	32
2.2.2 Multimodal Design Iteration	33
2.2.3 Rapid Prototyping Platforms and Design Libraries	34
2.3 Code Generation and Refinement	35
2.3.1 High-Level Directives to Autonomous Agents	35
2.3.2 Agent Modes for Complex Multi-File Transformations.....	36
2.3.3 From Draft to Done: Making AI-Generated Code Shippable	37
2.3.4 Quality, Security, and Performance Guardrails	38
2.4 Testing and Debugging AI-Generated Code	38
2.4.1 Iterative Debugging Loops and Automated Quality Gates.....	38
2.4.2 Test-Driven Development (TDD) with AI	39
2.4.3 Prioritizing Prototype Interaction	40
2.5 Version Control and Collaboration with AI	41
2.5.1 Git Integration, Checkpointing, and Parallel Experiments	41

2.5.2 Standardized Authorship Labeling	42
2.5.3 Collaboration Etiquette with AI and Humans	43
2.6 Summary and Takeaways	44
Review Questions.....	46
Multiple-Choice Questions	46
Short-Answer Conceptual Questions	48
Applied Project-Scenario Questions	49
Open-Ended Discussion Questions	50
Practical Lab Tasks	51
Chapter 3: Beyond Vibe Coding: Agentic Coding, Security, and Scalability	54
3.1 Vibe Coding vs. Agentic Coding: A Deep Dive	54
3.1.1 Philosophical Distinction	54
3.1.2 Execution Models & Architecture	55
3.1.3 The Hybrid Approach: Combining the Best of Both Worlds	57
3.2 Ensuring Code Quality and Architectural Coherence	58
3.2.1 Quality is Your Responsibility: When "Vibes" Aren't Enough for Production	58
3.2.2 Design-First Prompting (DFP)	60
3.2.3 Maintaining Architectural Stability: Building with a Strong Foundation	62
3.3 Security and Compliance in the AI Era	63
3.3.1 Secure-by-Design AI Coding Practices	63
3.3.2 Policy-as-Code (PaC): Your Automated Quality Checklist	64
3.3.3 Mitigating AI-Specific Risk: Your Final Security Checklist	66
3.4 Addressing Common Challenges.....	68
3.4.1 Model Hallucinations	68
3.4.2 Skill Atrophy and Over-Reliance	70
3.4.3 Knowledge Fragmentation: Keeping Your Project's Memory Intact	71
3.4.4 Context and Scope Drift: Keeping Your AI Focused	72
3.4.5 Token, Latency, and Cost Pressure: Managing Your AI's Resources	74
3.4.6 Flaky Signals: When Your Tests Lie	75
3.5 Best Practices for Effective Vibe Coding Adoption	76
3.5.1 People: Your Evolving Team and New Skills	76
3.5.2 Process: Building a Standard Way of Working	78
3.5.3 The Platform: Your AI-Powered Toolbox	79

3.5.4 Policy: Your Project's Digital Rulebook	80
3.5.6 Organizational Adoption: Managing the Change	82
3.6 Summary and Takeaways	83
Review Questions.....	85
Multiple-Choice Questions	85
Short-Answer Conceptual Questions	87
Applied Project-Scenario Questions	88
Discussion Questions	89
Practical Lab Tasks	90
Chapter 4: AI-Assisted Specification-Driven Development: An Overview	94
4.1. Introduction: Taming the Generative AI Chaos	94
4.1.1 The Vibe and Agentic Coder's Dilemma	94
4.1.2 The Blueprint for Vibe Coders.....	95
4.1.3 SDD Key Aspects	95
4.1.4 SDD with GenAI: Tools and Platforms.....	97
4.1.5 Slash Commands: The More Precise Language of Control	99
4.2. Specification Framework in the GenAI Era.....	101
4.2.1 Defining the Living Specification	101
4.2.2 Anatomy of an AI-Powered Spec	101
4.2.3 The New Engineer's Role.....	103
4.3. The SDD Workflow: From Idea to Production-Ready Code	103
4.3.1 Specify with a Vision.....	103
4.3.2 Plan and Refine with the Agent	104
4.3.3 Task Breakdown and Implementation	104
4.3.4 Review, Test, and Integrate	105
4.4. Practical Example: Building a "Vibe" App with SDD.....	105
4.4.1 The Project: VibeTunes	105
4.4.2 The Result.....	107
4.5 Benefits of Specification-Driven Development in the GenAI Era	107
4.5.1 Key Advantages	107
4.6 Challenges and Limitations of SDD with GenAI	108
4.6.1 Practical and Technical Limitations.....	109
4.6.2 Ethical and Societal Considerations.....	109

4.6.3 Overcoming the Challenges: A Hybrid and Adaptive Approach	110
4.7. SDD Best Practices for Advanced Vibe Coders	111
4.7.1 Context Engineering: Going Deeper	111
4.7.2 Agent Rules: Explicit Guardrails	111
4.7.3 Using Different AI Models	112
4.7.4 Version Control Integration	112
4.7.5 AI Model Drift Considerations	113
4.7.6 Collaboration in Team-Based AI Coding	113
4.8. Conclusion: A New Era of Engineering Excellence	114
4.8.1 The Mindset Shift	114
4.8.2 The Future is Hybrid	114
Review Questions	116
Multiple-Choice Questions	116
Short-Answer Conceptual Questions	118
Applied Project-Scenario Questions	119
Discussion Questions	120
Practical Lab Tasks	121
Chapter 5: The Executable Blueprint: Spec-Driven Development with GitHub Spec Kit	124
5.1 Introduction to SDD Platforms and GitHub SpecKit	124
5.1.1 Choosing the Right Tool: A Comparative Overview	125
5.2 GitHub Spec Kit Overview	127
5.2.1 The Essential Tools and Commands	127
5.2.2 Platform Integration	129
5.3 The Complete SpecKit Workflow: A Structured Development Protocol	130
5.3.1 Defining the Constitution	130
5.3.2 Specification Generation	131
5.3.3 Clarification and Risk Mitigation	132
5.3.4 Technical Planning	133
5.3.5 Task Decomposition	134
5.3.6 Analysis and Verification	135
5.3.7 Implementation Execution	135
5.4 Case Study: Building a Verified Application Step-by-Step	136
Phase 4: Tasks (The Action Plan)	139

Phase 5: Implement and Review (Execution).....	140
5.5 Scaling SDD with Advanced AI Integrations.....	141
5.5.1 Enforcing Guardrails and Quality	141
5.5.2 Specifications as Living Artifacts	141
5.5.3 Hybrid Approach and Practical Flexibility.....	142
5.5.4 Recap of Scaling SDD	142
5.6 Ethics Considerations and the Future Outlook	143
5.6.1 Ethical Imperatives in AI-Driven Development	143
5.6.2 The Future Outlook.....	143
5.7 Summary and Takeaways	144
5.7.1 Key Takeaways.....	145
Review Questions.....	147
Multiple-Choice Questions	147
Short-Answer Conceptual Questions	149
Applied Project-Scenario Questions	150
Discussion Questions	151
Practical Lab Tasks	152
Chapter 6: The Future of AI-Powered Software Development: Beyond Vibe Coding.....	156
6.1 The Evolving Landscape of Software Development	156
6.1.1 From Vibe Coding to Agentic Coding	157
6.1.2 Specification-Driven Development as the Framework of Trust	157
6.1.3 Future of AI-Human Collaboration: Deep Hybrid Workflows	157
6.1.4 Adaptive "Vibe-Aware" Agents	158
6.1.5 Continuous Model Fine-Tuning in IDEs	159
6.1.6 Context Engineering: The Cognitive Infrastructure	159
6.2 Transforming Software Engineering Paradigms: From SDLC to Agentic Systems 160	
6.2.1 SDLC Model → AI-Augmented Precision	160
6.2.2 Agile and Iterative Development → AI-Collaborative Agility.....	161
6.2.3 Design Patterns → Context-Aware Pattern Synthesis.....	161
6.2.4 Component-based Development → Automated Creation and Management....	162
6.2.5 DevOps and CI/CD → Autonomous Continuous Delivery	162
6.2.6 Distributed Systems → Intelligent Scalability and Security	163

6.2.7 From Methodologies to Ecosystems	163
6.3 Implications for Education and Developer Skill Evolution	164
6.3.1 Redefining Programming Skills: From Syntax to Computational Thinking and Prompting.....	164
6.3.2 Workforce Implications.....	164
6.3.3 Guarding Against Skill Atrophy	165
6.3.4 AI-Native Learning Environments and Curricula	166
6.4 Ethical Considerations and Governance	166
6.4.1 Transparency and Provenance Tracking: The SDD Audit Trail	167
6.4.2 Intellectual Property and Security.....	167
6.4.3 Explainable AI (XAI)	168
6.4.4 Ethical Guardrails and Continuous Verification	168
6.5 The Convergent Future: Toward Agentic Engineering	169
6.5.1 Core Characteristics of Agentic Engineering.....	169
6.5.2 A Vision Forward.....	170
6.6 Conclusion: The Inclusive Era of AI-Assisted Tools.....	170
Review Questions.....	172
Multiple-Choice Questions	172
Short-Answer Conceptual Questions	174
Applied Project-Scenario Questions	175
Open-Ended Discussion Questions	177
Practical Lab Tasks	178
Appendix A: The AI Coding Ecosystem: Tools and Agentic IDEs	183
A.1 Categories of AI Coding Tools	183
A.1.1 Intelligent IDEs/Agentic IDEs/Code Editors	183
A.1.2 Intelligent IDE vs Agentic IDE: A Shift in Control.....	184
A.1.3 Browser-Based Environments.....	184
A.1.4 CLI Agents (Command Line Interface)	184
A.1.5 Task Management Systems	184
A.2 Deep Dive into Prominent Vibe Coding Tools	185
A.2.1 Intelligent/Agentic IDEs and Code Editors	185
A.2.1.1 GitHub Copilot.....	185
A.2.1.2 Cursor	186

A.2.1.3 Kiro	187
A.2.1.4 Windsurf and Augment Code	188
A.2.1.5 Zed	188
A.2.2 Browser-Based Environments	188
A.2.2.1 Replit	188
A.2.2.2 Lovable	189
A.2.2.3 Firebase Studio	190
A.2.2.4 Softgen	190
A.2.2.5 v0 by Vercel	191
A.2.3 CLI Agents and Utilities	191
A.2.3.1 Claude Code	192
A.2.3.2 Gemini CLI	192
A.2.3.3 OpenAI Codex	193
A.2.3.4 Amazon Q Developer CLI	194
A.2.3.5 CodeRabbit CLI	195
A.3 AI Code Refactoring Tools	195
A.3.1. DocuWriter.ai	196
A.3.2 ZenCoder	196
A.4 Tools for Specification-Driven Development	197
A.4.1 GitHub Spec Kit	197
A.4.2 Fission AI OpenSpec	198
A.4.3 Amazon Kiro	198
A.4.4 BMAD (Method)	198
A.5 Summary and Takeaways	199
Appendix B: Hands-on Vibe Coding Projects	200
B.1 Building a Personal Portfolio Website	200
B.1.1 Project at a Glance	200
B.1.2 Step-by-Step Vibe Coding Workflow	200
B.1.3 Optional Enhancements and Deployment	202
B.2 Generating a Startup Landing Page	202
B.2.1 Project at a Glance	202
B.2.2 Step-by-Step Vibe Coding Process	203
B.3 Full-Stack Application with Authentication and Database	204

B.3.1 Project at a Glance	204
B.3.2 Step-by-Step Professional Vibe Coding Workflow.....	205
B.4 Automating Daily Email Reports with Advanced Error Handling	207
B.4.1 Project at a Glance	207
B.4.2 Step-by-Step Vibe Coding Workflow	208
B.4.3 Debugging Strategies: Trusting the Logs.....	209
B.4.4 Deployment Guidance: Making it Daily	209
B.5 Agentic Refactoring: Upgrading a Legacy Project	210
B.5.1 Project at a Glance	210
B.5.2 Step-by-Step Agentic Workflow: Human Oversight is Key.....	210
B.6 Building a Custom AI Agent.....	212
B.6.1 Project at a Glance	212
B.6.2 Minimal Architecture (The Agent Team)	213
B.6.3 Step-by-Step Agent Orchestration Workflow	213
B.7 Website Cloning and Design Conversion	215
B.7.1 Project at a Glance	215
B.7.2 Step-by-Step Agentic Workflow: Structured, Safe, and Ethical	215
B.8 Revamping UI Design for an Existing Application (Multimodal and Agentic)	217
B.8.1 Project at a Glance	218
B.8.2 Step-by-Step Multimodal + Agentic Workflow	218
B.9 Summary	220
Appendix C: Developing a Social Media Feature with Vibe Coding, Agentic Coding, and the Hybrid Approach.....	221
C.1 At a Glance: Comparing the Three Approaches	221
C.2 Vibe Coding Workflow: Rapid Creation through Conversation	221
C.3 Agentic Coding Workflow: Structured Autonomy and Verification.....	222
C.4 Hybrid Workflow: Balancing Creativity and Discipline	223
C.5 Reflection: Choosing the Right Approach	225
C.6 Summary: The Developer's New Role	225

Preface: From Speed to Structure — The New Era of AI-Powered Software Development

The story of modern software development is, in many ways, the story of speed. With the rise of **Vibe Coding**, developers will no longer wrestle with rigid syntax or struggle to debug line by line. Instead, they simply describe their intent in plain English, AI assistants and coding agents could translate intent into prototypes within minutes. This shift makes software development not only faster, but also more creative, more accessible, and far more experimental.

But with that speed comes chaos. As projects grow larger, the lack of structure in Vibe Coding makes it difficult to ensure quality, consistency, and accountability. Code is generated faster than it can be verified, and creative freedom often collides with the need for maintainability and security. The challenge becomes clear: **How can we harness the speed and creativity of AI without sacrificing reliability and engineering rigor?**

The answer lies in a new generation of methodologies designed for the era of AI-powered software development, combining the spontaneity of Vibe Coding with AI agents and the discipline of software engineering. Among them, **AI-Assisted Specification-Driven Development (SDD)** has emerged as one of the most promising. It provides a structured framework that translates intent into verifiable specifications — requirements, designs, and tasks — guiding AI agents through a more reliable, reproducible process.

Today, SDD exists in several evolving forms. **BMAD**, the "Breakthrough Method for Agile AI-Driven Development," is an open-source framework that combines agile software development with AI agents to create a more structured and efficient workflow. **Fission AI's OpenSpec** is a lightweight, spec-driven development tool that provides a structured workflow to ensure the AI agents follow a clear, step-by-step plan, reducing uncertainty and preventing requirements from getting lost along the way. **Amazon Kiro** represents a full-fledged agentic environment with autonomous implementation and testing. And **GitHub Spec Kit**, one of the most widely adopted, brings this methodology to everyday developers through an open, command-driven workflow that balances automation with human oversight.

These systems share a common goal: to bring **structure, verification, and governance** into AI-assisted software creation. Just as traditional software engineering has matured over generations and through various foundational approaches, including Waterfall, SDLC, Agile, Design Patterns, Component-Based Development, DevOps,

and CI/CD, AI-powered software development will also diversify into complementary approaches. In the future, Specification-Driven Development may not be the only methodology, but it will remain a cornerstone of this new software engineering landscape.

This book traces that evolution. It begins with the creative freedom of **Vibe Coding**, transitions into the coordinated intelligence of **Agentic Systems**, and culminates with a detailed exploration of **AI-assisted Specification-Driven Development**—with special focus on GitHub Spec Kit as a practical, real-world example.

The goal is simple yet ambitious: to show how **speed and structure, creativity and control**, and **human and machine intelligence** can finally work together in harmony. As we stand at the dawn of this new era of software development, one truth has become clear—the future of software engineering belongs not to code alone, but to collaboration between human intent and intelligent automation.