**AI for Software Managers**

**Course Workbook**

**January 20-22, 2026**  
**Digital Campus Vorarlberg**  
**Instructor: Frank Blau**

**Welcome**

This workbook is your guide and workspace for the AI for Software Managers workshop. It contains frameworks, templates, and exercises for each evening session. Use the blank spaces to take notes, complete exercises, and develop your organization's AI strategy.

**How to Use This Workbook**

* Each section corresponds to a workshop session
* Complete exercises during the designated workshop time
* Use the templates as starting points for your organization
* Add your own notes in the margins and blank spaces
* Bring this workbook to all three evening sessions

**Contact Information**

**Instructor:** Frank Blau  
**Email:** fjblau@gmail.com  
**Phone:** +43 663 0660 1372

**Table of Contents**

**Evening 1: Signal vs. Noise - Cutting Through the AI Hype**

* Opening Assessment: Where Is Your Organization Today?
* The Reality Check Framework
* AI Tool Evaluation Matrix
* ROI Calculator Template
* Pilot Program Playbook
* Risk Assessment Templates
* Stakeholder Alignment Guide
* Exercise Worksheets

**Evening 2: Leading Humans in the Age of Copilots**

* Homework Review Worksheet
* The AI Anxiety Spectrum
* Skills Evolution Map
* Performance Review Framework
* Code Review 2.0 Guidelines
* Team Health Dashboard
* 1:1 Conversation Templates
* Career Development Planning

**Evening 3: The New Rules of Software Delivery**

* Quality Gates Checklist
* AI Code Governance Framework
* Vendor Risk Assessment
* Security Considerations Checklist
* Competitive Advantage Planning
* 18-Month Strategic Roadmap
* Implementation Planning Templates

**Evening 1: Signal vs. Noise - Cutting Through the AI Hype**

**Opening Assessment: Where Is Your Organization Today?**

Rate your organization on each dimension (1 = Just Starting, 5 = Advanced)

**AI Awareness**

* Management understanding of AI capabilities: ☐1 ☐2 ☐3 ☐4 ☐5
* Developer familiarity with AI tools: ☐1 ☐2 ☐3 ☐4 ☐5
* Executive support for AI adoption: ☐1 ☐2 ☐3 ☐4 ☐5

**Current AI Usage**

* Number of developers using AI tools regularly: \_\_\_\_\_\_\_
* AI tools currently in use: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Budget allocated for AI tools: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Organizational Readiness**

* Clear policies for AI tool usage: ☐Yes ☐No ☐In Development
* Code review processes for AI-generated code: ☐Yes ☐No ☐In Development
* Training programs for AI tools: ☐Yes ☐No ☐Planned

**Challenges You're Facing**

List your top three AI-related challenges:

**Goals for This Workshop**

What do you hope to accomplish?

**The Reality Check Framework**

**Understanding AI Capabilities vs. Limitations**

**What AI Tools Can Do Well:**

* Generate code from clear specifications
* Suggest completions for repetitive patterns
* Explain existing code
* Translate between programming languages
* Generate test cases
* Document code automatically
* Find common bugs and security issues

**What AI Tools Cannot Do (Yet):**

* Understand complex business requirements without context
* Make architectural decisions
* Evaluate tradeoffs between different approaches
* Debug complex system interactions
* Understand your specific codebase without examples
* Replace human judgment and creativity
* Maintain context across large codebases

**Common AI Tool Categories**

**Code Completion and Generation**

* Examples: GitHub Copilot, Amazon CodeWhisperer, Tabnine
* Best for: Writing boilerplate code, standard patterns
* Limitations: May not understand your specific architecture

**Code Analysis and Review**

* Examples: SonarQube with AI, DeepCode
* Best for: Finding security issues, code quality problems
* Limitations: Cannot evaluate business logic correctness

**Documentation and Explanation**

* Examples: ChatGPT, Claude, specialized documentation tools
* Best for: Explaining code, generating documentation
* Limitations: May misunderstand complex or unusual code

**Testing**

* Examples: Test generation tools, AI-powered test frameworks
* Best for: Creating unit tests, edge case identification
* Limitations: Cannot understand intended behavior without guidance

**AI Tool Evaluation Matrix**

Use this matrix to evaluate any AI tool you are considering.

**Tool Being Evaluated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Evaluation Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Evaluated By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Category 1: Capabilities (Weight: 25%)**

| **Criterion** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Addresses our specific needs | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Works with our tech stack | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Integration with existing tools | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Quality of output | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Ease of use | ☐1 ☐2 ☐3 ☐4 ☐5 |  |

**Category Score:** \_\_\_\_\_ / 25

**Category 2: Cost and Value (Weight: 20%)**

| **Criterion** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Pricing clarity | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Cost per developer reasonable | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Clear ROI potential | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Free trial available | ☐1 ☐2 ☐3 ☐4 ☐5 |  |

**Category Score:** \_\_\_\_\_ / 20

**Category 3: Security and Privacy (Weight: 25%)**

| **Criterion** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Data privacy controls | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Code confidentiality | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Compliance certifications | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| On-premise option available | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Clear terms of service | ☐1 ☐2 ☐3 ☐4 ☐5 |  |

**Category Score:** \_\_\_\_\_ / 25

**Category 4: Vendor Stability (Weight: 15%)**

| **Criterion** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Company track record | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Customer base size | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Financial stability | ☐1 ☐2 ☐3 ☐4 ☐5 |  |

**Category Score:** \_\_\_\_\_ / 15

**Category 5: Support and Documentation (Weight: 15%)**

| **Criterion** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Documentation quality | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Support responsiveness | ☐1 ☐2 ☐3 ☐4 ☐5 |  |
| Community and resources | ☐1 ☐2 ☐3 ☐4 ☐5 |  |

**Category Score:** \_\_\_\_\_ / 15

**Total Score: \_\_\_\_\_ / 100**

**Decision Criteria**

* 80-100: Strong candidate, proceed with pilot
* 60-79: Potential fit, needs more evaluation
* 40-59: Significant concerns, evaluate alternatives
* Below 40: Not recommended

**Next Steps**

☐ Schedule pilot program  
☐ Request additional information  
☐ Evaluate alternative tools  
☐ Reject and document reasons

**ROI Calculator Template**

**Calculating the Return on Investment for AI Tools**

**Step 1: Calculate Costs**

**Tool Costs (Annual)**

* License fees per developer: € \_\_\_\_\_\_\_
* Number of developers: \_\_\_\_\_\_\_
* Total license cost: € \_\_\_\_\_\_\_
* Infrastructure costs: € \_\_\_\_\_\_\_
* Training costs: € \_\_\_\_\_\_\_
* Implementation time (hours × cost per hour): € \_\_\_\_\_\_\_

**Total Annual Cost:** € \_\_\_\_\_\_\_

**Step 2: Calculate Expected Benefits**

**Time Savings**

* Average hours saved per developer per week: \_\_\_\_\_\_\_
* Number of developers: \_\_\_\_\_\_\_
* Weeks per year: \_\_\_\_\_\_\_
* Total hours saved: \_\_\_\_\_\_\_
* Average developer hourly cost: € \_\_\_\_\_\_\_
* Total labor savings: € \_\_\_\_\_\_\_

**Quality Improvements**

* Estimated reduction in bugs: \_\_\_\_\_\_\_%
* Average cost per bug: € \_\_\_\_\_\_\_
* Number of bugs per year: \_\_\_\_\_\_\_
* Bug reduction savings: € \_\_\_\_\_\_\_

**Faster Delivery**

* Estimated project acceleration: \_\_\_\_\_\_\_%
* Value of faster time to market: € \_\_\_\_\_\_\_

**Total Annual Benefits:** € \_\_\_\_\_\_\_

**Step 3: Calculate ROI**

ROI = (Total Benefits - Total Costs) / Total Costs × 100

**ROI:** \_\_\_\_\_\_\_%

**Payback Period:** \_\_\_\_\_\_\_ months

**Reality Check Questions**

Before finalizing your ROI calculation, consider:

* Are the time savings realistic? (Consider learning curve)
* Will all developers achieve the same productivity gains?
* Have you accounted for productivity loss during adoption?
* Are there hidden costs you have not included?
* Is the quality improvement estimate conservative?

**Notes and Adjustments**

**The Pilot Program Playbook**

**Designing an Effective AI Tool Pilot**

**Pilot Objectives**

What specific questions should this pilot answer?

**Pilot Parameters**

* Duration: \_\_\_\_\_\_\_ weeks (Recommended: 4-8 weeks)
* Team size: \_\_\_\_\_\_\_ developers
* Projects included: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Success metrics: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Team Selection Criteria**

Choose pilot participants who are:

* Open to trying new tools
* Willing to provide honest feedback
* Working on representative projects
* Mix of experience levels
* Strong communicators

**Pilot Team Members**

| **Name** | **Role** | **Experience Level** | **Project** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Measurement Plan**

What will you measure during the pilot?

**Quantitative Metrics:**

* Lines of code written per day
* Time to complete tasks
* Number of code reviews required
* Bug rates
* Test coverage

**Qualitative Metrics:**

* Developer satisfaction
* Ease of use
* Quality of generated code
* Integration with workflow
* Learning curve

**Data Collection Method**

How will you gather this data?

☐ Developer surveys (weekly)  
☐ Code repository analysis  
☐ Time tracking  
☐ Bug tracking system reports  
☐ One-on-one interviews  
☐ Team retrospectives

**Pilot Timeline**

| **Week** | **Activities** | **Responsible** |
| --- | --- | --- |
| 1 | Setup and training |  |
| 2-7 | Active usage and monitoring |  |
| 8 | Feedback collection and analysis |  |
| 9 | Decision and recommendations |  |

**Success Criteria**

Define clear criteria for continuing with the tool:

Minimum acceptable results:

* Productivity increase: \_\_\_\_\_\_\_%
* Developer satisfaction: \_\_\_\_\_\_\_ out of 5
* Code quality maintained or improved: ☐Yes ☐No
* ROI potential: \_\_\_\_\_\_\_%

**Pilot Budget**

* Tool costs: € \_\_\_\_\_\_\_
* Training time: € \_\_\_\_\_\_\_
* Monitoring and analysis: € \_\_\_\_\_\_\_
* Total pilot cost: € \_\_\_\_\_\_\_

**Risk Assessment Templates**

**Technical Risks**

| **Risk** | **Likelihood (1-5)** | **Impact (1-5)** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| Tool does not integrate with existing systems |  |  |  |
| Generated code quality is poor |  |  |  |
| Tool requires too much configuration |  |  |  |
| Performance impact on development environment |  |  |  |
| Tool lacks features for our tech stack |  |  |  |

**Financial Risks**

| **Risk** | **Likelihood (1-5)** | **Impact (1-5)** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| ROI does not materialize |  |  |  |
| Unexpected cost increases |  |  |  |
| Vendor changes pricing model |  |  |  |
| Required infrastructure upgrades |  |  |  |

**Organizational Risks**

| **Risk** | **Likelihood (1-5)** | **Impact (1-5)** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| Developer resistance to adoption |  |  |  |
| Loss of critical development skills |  |  |  |
| Over-reliance on AI tools |  |  |  |
| Uneven adoption across teams |  |  |  |
| Management unrealistic expectations |  |  |  |

**Security and Privacy Risks**

| **Risk** | **Likelihood (1-5)** | **Impact (1-5)** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| Code leaked to training data |  |  |  |
| Intellectual property concerns |  |  |  |
| Compliance violations |  |  |  |
| Security vulnerabilities in generated code |  |  |  |
| Lack of audit trail |  |  |  |

**Stakeholder Alignment Guide**

**Identifying Key Stakeholders**

List everyone who needs to be informed or consulted:

**Executive Level**

* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_
* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Management Level**

* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_
* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Developer Level**

* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_
* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Other Stakeholders** (Legal, Security, Procurement, etc.)

* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_
* Name: \_\_\_\_\_\_\_\_\_\_\_\_\_ Role: \_\_\_\_\_\_\_\_\_\_\_\_\_ Concern: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Stakeholder Communication Plan**

**For Executives:**

Key messages:

* Competitive advantage potential
* ROI and cost considerations
* Risk mitigation approach
* Timeline and milestones

Preferred format: ☐Presentation ☐One-pager ☐Email ☐Meeting

**For Developers:**

Key messages:

* How this helps them (not replaces them)
* Training and support available
* Pilot approach (not forced adoption)
* Feedback mechanisms

Preferred format: ☐Team meeting ☐Demo ☐Written guide ☐Q&A session

**For Legal/Security:**

Key messages:

* Data privacy measures
* Compliance considerations
* Security controls
* Vendor evaluation criteria

Preferred format: ☐Detailed report ☐Meeting ☐Policy document

**Addressing Common Objections**

**"We cannot afford it"**

Your response:

**"Our developers will lose their skills"**

Your response:

**"AI-generated code is not secure"**

Your response:

**"This is just a fad"**

Your response:

**Exercise: Evaluate an AI Tool**

**Tool Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Use the AI Tool Evaluation Matrix (page 8) to score the tool you are considering.

Work in small groups or individually. You have 30 minutes.

**After completing the evaluation:**

1. What was your total score? \_\_\_\_\_\_\_
2. What were the strongest points?
3. What were the weakest points?
4. Would you recommend this tool for a pilot? ☐Yes ☐No
5. What additional information do you need?

**Day 1 Homework Assignment**

**Assignment:** Evaluate one AI tool your team is considering or currently using.

**Steps:**

1. Choose an AI tool relevant to your work
2. Complete the AI Tool Evaluation Matrix (page 8)
3. Calculate a basic ROI using the template (page 11)
4. Bring your completed evaluation to Day 2

**Tool I will evaluate:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Why this tool:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Evening 2: Leading Humans in the Age of Copilots**

**Homework Review Worksheet**

Share your AI tool evaluation with your group.

**Your tool:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Your total score:** \_\_\_\_\_\_\_

**Key findings:**

Strengths:

Weaknesses:

**ROI estimate:** \_\_\_\_\_\_\_%

**Would you proceed with this tool?** ☐Yes ☐No ☐Maybe

**Patterns from Group Discussion**

What patterns did you notice across different tool evaluations?

**The AI Anxiety Spectrum**

**Understanding Different Reactions to AI**

Developers respond to AI tools in different ways. Understanding these reactions helps you manage your team effectively.

**The Resister** (10-20% of developers)

* Refuses to try AI tools
* Sees AI as a threat to their job
* Believes AI cannot match human quality
* May feel their expertise is being devalued

**Management approach:**

* Acknowledge their concerns as valid
* Emphasize AI as a tool, not a replacement
* Show examples of AI limitations
* Give them time and space
* Do not force adoption immediately

**The Skeptic** (30-40% of developers)

* Willing to try but doubtful
* Concerned about quality and security
* Needs proof before committing
* Will adopt if convinced

**Management approach:**

* Provide data and evidence
* Start with low-risk projects
* Address specific concerns directly
* Let them discover benefits themselves

**The Pragmatist** (30-40% of developers)

* Sees AI as another tool
* Uses it when helpful, ignores when not
* Focused on getting work done
* Balanced perspective

**Management approach:**

* Support their experimentation
* Share their experiences with the team
* Use them as pilot participants
* Learn from their balanced approach

**The Enthusiast** (10-20% of developers)

* Excited about AI possibilities
* Early adopter
* May over-rely on AI
* Can be blind to limitations

**Management approach:**

* Channel their enthusiasm productively
* Watch for over-dependence
* Ensure they maintain critical thinking
* Use them to help train others

**Your Team Profile**

Map your team members to these categories:

**Resisters:**

**Skeptics:**

**Pragmatists:**

**Enthusiasts:**

**Individual Management Plans**

For each person who needs special attention, create a plan:

**Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Category:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Specific concerns:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Your approach:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Timeline:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Skills Evolution Map**

**What Skills Are Changing?**

**Skills That Remain Critical** (Analog Awareness)

These human skills become more valuable with AI:

* System design and architecture
* Understanding business requirements
* Problem decomposition
* Code review and quality judgment
* Security thinking
* Performance optimization
* Debugging complex issues
* Team communication
* Mentoring others
* Strategic technical decisions

**Skills That Are Changing**

These skills need to adapt:

* Code writing (less manual typing, more guiding AI)
* Syntax memorization (less important)
* Boilerplate code creation (largely automated)
* Documentation writing (AI-assisted)
* Test writing (AI can help generate)

**New Skills Needed** (Digital Competency)

Your team needs to develop:

* Prompt engineering (how to ask AI effectively)
* AI output evaluation (spotting errors quickly)
* Tool selection (choosing right AI tool for task)
* Hybrid workflows (combining human and AI work)
* AI limitations awareness (knowing when not to use AI)

**Team Skills Assessment**

Rate your team's current capability in each area:

| **Skill Area** | **Current Level (1-5)** | **Target Level** | **Gap** |
| --- | --- | --- | --- |
| System architecture |  |  |  |
| Requirements analysis |  |  |  |
| Code review quality |  |  |  |
| Security awareness |  |  |  |
| Prompt engineering |  |  |  |
| AI output evaluation |  |  |  |
| Tool selection |  |  |  |

**Training and Development Plan**

Based on your skills gap analysis:

**Immediate training needs** (Next 3 months):

**Medium-term development** (3-6 months):

**Long-term growth** (6-12 months):

**Training resources needed:**

☐ External training courses  
☐ Internal workshops  
☐ Online learning platforms  
☐ Mentoring programs  
☐ Conference attendance  
☐ Certifications

**Budget required:** € \_\_\_\_\_\_\_

**Performance Review Framework**

**Adapting Performance Evaluation for AI-Augmented Work**

Traditional metrics like lines of code written become less meaningful when AI generates code. Here are new ways to evaluate performance:

**New Performance Indicators**

**Quality Over Quantity**

* Code that passes review on first submission
* Number of bugs in production
* Test coverage and quality
* Documentation clarity
* Architecture decisions quality

**Problem Solving**

* Ability to decompose complex problems
* Solution creativity and elegance
* Understanding of business context
* Speed of debugging

**AI Tool Mastery**

* Effective use of AI tools
* Ability to evaluate AI output critically
* Knowing when not to use AI
* Teaching others effective AI use

**Collaboration**

* Code review quality and helpfulness
* Knowledge sharing
* Mentoring effectiveness
* Team communication

**Continuous Learning**

* Adaptation to new tools
* Skill development
* Staying current with technology
* Sharing learnings with team

**Performance Review Template**

**Developer:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Review Period:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reviewer:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Code Quality and Delivery**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**2. Problem Solving and Architecture**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**3. AI Tool Effectiveness**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**4. Critical Thinking**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**5. Collaboration and Mentoring**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**6. Continuous Learning**

Rate: ☐1 ☐2 ☐3 ☐4 ☐5

Evidence:

**Overall Assessment:**

Strengths:

Areas for development:

Goals for next period:

**Code Review 2.0 Guidelines**

**Reviewing AI-Generated Code**

When code is written with AI assistance, code review becomes even more critical. Here is how to adapt your review process:

**Standard Review Checklist**

☐ **Functionality:** Does the code do what it is supposed to do?  
☐ **Correctness:** Is the logic correct?  
☐ **Edge Cases:** Are boundary conditions handled?  
☐ **Error Handling:** Are errors caught and managed properly?  
☐ **Testing:** Are there adequate tests?  
☐ **Documentation:** Is the code well documented?  
☐ **Style:** Does it follow team standards?  
☐ **Security:** Are there security vulnerabilities?  
☐ **Performance:** Is it efficient?  
☐ **Maintainability:** Can others understand and modify it?

**Additional Checks for AI-Generated Code**

☐ **Understanding:** Does the developer understand the code?  
☐ **Over-Engineering:** Is the solution unnecessarily complex?  
☐ **Context Awareness:** Does it fit the existing codebase?  
☐ **Hidden Assumptions:** Are there unstated assumptions?  
☐ **License Issues:** Could there be license violations?  
☐ **Outdated Patterns:** Is it using old approaches?  
☐ **Copy-Paste Errors:** Are there duplicated or inconsistent sections?

**Code Review Questions to Ask**

**For the Developer:**

1. "Can you explain how this code works?"
2. "Did you write this or did AI generate it?"
3. "Did you test all the edge cases?"
4. "Why did you choose this approach?"
5. "What alternatives did you consider?"

**Red Flags:**

* Developer cannot explain the code
* Code is overly complex for the problem
* Style does not match the rest of the codebase
* Suspicious comments or variable names
* Lack of error handling
* No tests provided

**Team Code Review Standards**

Update your code review standards for AI-assisted development:

**Our team standards:**

**AI-specific requirements:**

**Team Health Dashboard**

**Monitoring AI Adoption Impact**

Use this dashboard monthly to track how AI adoption is affecting your team.

**Month:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Team Metrics**

| **Metric** | **Current** | **Last Month** | **Trend** |
| --- | --- | --- | --- |
| Team morale (1-5 scale) |  |  |  |
| Average task completion time |  |  |  |
| Code review cycles needed |  |  |  |
| Production bugs |  |  |  |
| Developer satisfaction (1-5) |  |  |  |
| AI tool usage rate |  |  |  |

**Individual AI Adoption**

| **Developer** | **Using AI?** | **Comfort Level (1-5)** | **Concerns** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Qualitative Indicators**

How many developers in each category?

* Resistant to AI: \_\_\_\_\_\_\_
* Skeptical but trying: \_\_\_\_\_\_\_
* Comfortable using AI: \_\_\_\_\_\_\_
* Over-reliant on AI: \_\_\_\_\_\_\_

**Team Concerns This Month**

**Actions Taken This Month**

**Actions Needed Next Month**

**1:1 Conversation Templates**

**Having Difficult Conversations About AI**

**Conversation 1: The Resistant Developer**

**Opening:** "I have noticed you have not been using the new AI tools we introduced. I wanted to understand your perspective."

**Listening for:**

* Specific concerns about job security
* Quality worries
* Skill preservation fears
* Past negative experiences

**Key messages to convey:**

* Their expertise is valued and needed
* AI is a tool, not a replacement
* No one is forced to use AI
* We want their honest feedback

**Closing:** "What would make you more comfortable exploring these tools, even just to evaluate them?"

**Your notes for this conversation:**

**Conversation 2: The Over-Reliant Developer**

**Opening:** "I wanted to talk about how you are using AI tools in your work."

**Listening for:**

* Understanding level of the code they submit
* Testing and validation practices
* Time saved vs. time spent fixing issues

**Key messages to convey:**

* AI is a productivity tool, not a thinking replacement
* Understanding code is critical
* Balance is important
* We value quality over speed

**Closing:** "What is one way you can verify AI-generated code more thoroughly?"

**Your notes for this conversation:**

**Conversation 3: Career Development with AI**

**Opening:** "Let's talk about how AI tools are changing the skills you need to develop."

**Topics to cover:**

* Skills that remain valuable
* New skills to learn
* Career path options
* Growth opportunities

**Questions to ask:**

* Where do you see your career in 2-3 years?
* What skills do you want to develop?
* How can AI tools help you grow?
* What concerns do you have about your career?

**Your notes for this conversation:**

**Career Development Planning**

**Helping Your Team Grow in the AI Era**

**Developer:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Current Role:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Career Goals:**

**Strengths:**

**Development Areas:**

**AI-Related Skills Assessment**

| **Skill** | **Current (1-5)** | **Target (1-5)** | **Development Plan** |
| --- | --- | --- | --- |
| AI tool usage |  |  |  |
| Prompt engineering |  |  |  |
| Critical evaluation of AI output |  |  |  |
| Architecture and design |  |  |  |
| System thinking |  |  |  |
| Code review quality |  |  |  |

**6-Month Development Plan**

**Quarter 1:**

Goals:

Activities:

**Quarter 2:**

Goals:

Activities:

**Support Needed:**

☐ Training budget  
☐ Time for learning  
☐ Mentoring  
☐ Conference attendance  
☐ Certification  
☐ Project assignments

**Success Measures:**

**Day 2 Wrap-Up**

**Key Takeaways from Today:**

**Actions I Will Take:**

**Questions I Still Have:**

**Evening 3: The New Rules of Software Delivery**

**Quality Gates Reimagined**

**Adapting Your Development Process for AI**

Traditional quality gates need updating when AI generates code. Here is a new framework:

**Quality Gate 1: Requirements and Design**

**Before AI assistance:** ☐ Requirements are clear and documented  
☐ Design approach is reviewed  
☐ Architecture decisions are approved  
☐ Security considerations are identified  
☐ Performance requirements are defined

**Why this matters:** AI needs clear context to generate appropriate code.

**Quality Gate 2: Code Generation**

**During AI-assisted coding:** ☐ Developer understands the generated code  
☐ Code follows team standards  
☐ Appropriate tests are created  
☐ Documentation is updated  
☐ Security scan is clean

**Why this matters:** Prevents blind acceptance of AI output.

**Quality Gate 3: Code Review**

**Before merging:** ☐ Another developer reviews the code  
☐ All automated tests pass  
☐ Code review checklist is complete  
☐ No security vulnerabilities found  
☐ Performance is acceptable  
☐ Developer can explain the code

**Why this matters:** Catches issues that AI and the developer may miss.

**Quality Gate 4: Testing**

**Before deployment:** ☐ Unit tests pass  
☐ Integration tests pass  
☐ Manual testing completed  
☐ Edge cases verified  
☐ Performance testing done

**Why this matters:** AI-generated code may work for common cases but fail on edge cases.

**Quality Gate 5: Production**

**After deployment:** ☐ Monitoring is in place  
☐ Error rates are tracked  
☐ Performance metrics are monitored  
☐ User feedback is collected  
☐ Rollback plan is ready

**Why this matters:** Quickly identify issues in production.

**Your Quality Gates**

Document your team's quality gates:

**Gate 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Criteria:

Responsible: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Gate 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Criteria:

Responsible: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Gate 3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Criteria:

Responsible: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**AI Code Governance Framework**

**Establishing Policies That Work**

**Policy 1: AI Tool Approval**

Who can approve new AI tools?

☐ Individual developers  
☐ Team lead  
☐ Engineering manager  
☐ CTO  
☐ Security team  
☐ IT department

**Process for approving new tools:**

**Policy 2: Disclosure Requirements**

When must developers disclose AI usage?

☐ Never (AI use is always allowed)  
☐ In code comments  
☐ In commit messages  
☐ In code review  
☐ For substantial code sections only  
☐ Always

**Our policy:**

**Policy 3: Code Ownership**

Who owns AI-generated code?

☐ The company  
☐ The developer  
☐ The AI vendor  
☐ Unclear

**Legal review status:**

**Policy 4: Acceptable Use**

AI tools can be used for:

☐ Writing new code  
☐ Refactoring existing code  
☐ Writing tests  
☐ Documentation  
☐ Code review assistance  
☐ Learning and training  
☐ Architecture suggestions

AI tools cannot be used for:

☐ Copying proprietary code  
☐ Generating code for production without review  
☐ Security-critical code  
☐ Regulated functionality  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Policy 5: Data Privacy**

What code can be sent to AI tools?

☐ All code  
☐ Only non-proprietary code  
☐ Only code in public repositories  
☐ No code with customer data  
☐ No code with credentials or secrets  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Our data privacy rules:**

**Policy 6: Quality Standards**

AI-generated code must:

☐ Pass all automated tests  
☐ Be reviewed by another developer  
☐ Meet code coverage requirements  
☐ Follow team coding standards  
☐ Be understood by the submitting developer  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Policy 7: Security Requirements**

☐ All AI-generated code must pass security scanning  
☐ Security team reviews AI-generated authentication code  
☐ Penetration testing includes AI-generated code  
☐ Regular audits of AI tool usage  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Policy 8: Training and Onboarding**

New developers must:

☐ Complete AI tool training  
☐ Understand team AI policies  
☐ Sign acknowledgment of guidelines  
☐ Shadow experienced developer using AI  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Your Governance Document**

Compile your policies into a single document:

**AI Code Governance Policy**

**Version:** \_\_\_\_\_\_\_  
**Effective Date:** \_\_\_\_\_\_\_  
**Review Date:** \_\_\_\_\_\_\_  
**Owner:** \_\_\_\_\_\_\_

**Summary of key policies:**

**Vendor Risk Assessment**

**Evaluating and Managing Vendor Relationships**

**Vendor:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tool:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assessment Date:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Category 1: Vendor Stability**

| **Factor** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Years in business |  |  |
| Customer base size |  |  |
| Financial stability |  |  |
| Leadership team experience |  |  |
| Market position |  |  |

**Category 2: Product Maturity**

| **Factor** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Product age |  |  |
| Feature completeness |  |  |
| Release frequency |  |  |
| Bug fix responsiveness |  |  |
| Roadmap clarity |  |  |

**Category 3: Lock-in Risk**

| **Factor** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Data portability |  |  |
| API availability |  |  |
| Standard formats used |  |  |
| Integration dependencies |  |  |
| Alternative tools available |  |  |

**Category 4: Support and Documentation**

| **Factor** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Documentation quality |  |  |
| Support availability |  |  |
| Response time |  |  |
| Community size |  |  |
| Training resources |  |  |

**Category 5: Pricing Stability**

| **Factor** | **Rating (1-5)** | **Notes** |
| --- | --- | --- |
| Pricing model clarity |  |  |
| Price change history |  |  |
| Contract terms |  |  |
| Volume discount options |  |  |

**Risk Mitigation Strategies**

For each significant risk, plan a mitigation:

**Risk 1:**

**Mitigation:**

**Risk 2:**

**Mitigation:**

**Risk 3:**

**Mitigation:**

**Contract Requirements**

Include these terms in your contract:

☐ Data deletion upon termination  
☐ Export capability for all data  
☐ Price increase limits  
☐ Service level agreements  
☐ Liability limits  
☐ Privacy guarantees  
☐ Right to audit

**Security Considerations Checklist**

**Protecting Your Code and Data**

**Data Security**

☐ Understand what data the AI tool collects  
☐ Know where your data is stored  
☐ Verify data encryption in transit  
☐ Verify data encryption at rest  
☐ Review data retention policies  
☐ Confirm data deletion procedures  
☐ Check for data residency requirements  
☐ Review third-party data sharing policies

**Access Control**

☐ Implement least-privilege access  
☐ Use single sign-on (SSO) if available  
☐ Enable multi-factor authentication  
☐ Review user access regularly  
☐ Have process for removing access  
☐ Monitor for suspicious activity

**Code Security**

☐ Scan all AI-generated code for vulnerabilities  
☐ Review authentication and authorization code  
☐ Check for hardcoded credentials  
☐ Validate input handling  
☐ Review error messages for information leaks  
☐ Test for common vulnerabilities (SQL injection, XSS, etc.)

**Intellectual Property**

☐ Understand code ownership  
☐ Review AI tool training data policies  
☐ Protect proprietary algorithms  
☐ Check for license compliance  
☐ Document code sources  
☐ Review terms of service

**Compliance**

☐ Identify applicable regulations (GDPR, HIPAA, etc.)  
☐ Verify vendor compliance certifications  
☐ Review audit capabilities  
☐ Document AI tool usage for compliance  
☐ Train team on compliance requirements

**Incident Response**

☐ Plan for data breach scenarios  
☐ Know vendor notification procedures  
☐ Have rollback procedures ready  
☐ Document incident response steps  
☐ Test incident response plan

**Security Concerns for Your Organization**

List your specific security concerns:

**Mitigations needed:**

**Competitive Advantage Planning**

**Building Real Advantages Through AI**

AI tools are widely available, so using them does not automatically create competitive advantage. Here is how to build real advantages:

**Level 1: Tool Adoption** (No Advantage) Simply using AI tools like everyone else.

**Level 2: Effective Usage** (Temporary Advantage) Using AI tools better than competitors through training and process.

**Level 3: Custom Integration** (Medium-term Advantage) Integrating AI deeply into your workflows and tools.

**Level 4: Proprietary AI** (Long-term Advantage) Building AI capabilities specific to your domain and needs.

**Your Competitive Advantage Strategy**

**Current Level:** ☐1 ☐2 ☐3 ☐4

**Target Level:** ☐1 ☐2 ☐3 ☐4

**What makes your approach unique?**

**What advantages can you build?**

**Domain Knowledge Advantages:**

* Your specific industry expertise
* Proprietary processes
* Unique workflows
* Customer understanding

**How to leverage:**

**Technical Advantages:**

* Custom tooling
* Specialized infrastructure
* Unique architecture
* Performance optimizations

**How to leverage:**

**Team Advantages:**

* Skilled developers
* Effective processes
* Strong culture
* Fast learning

**How to leverage:**

**Data Advantages:**

* Proprietary datasets
* Customer insights
* Performance metrics
* Historical learnings

**How to leverage:**

**12-Month Advantage Building Plan**

**Months 1-3: Foundation**

Goals:

Actions:

**Months 4-6: Optimization**

Goals:

Actions:

**Months 7-9: Innovation**

Goals:

Actions:

**Months 10-12: Leadership**

Goals:

Actions:

**18-Month Strategic Roadmap**

**Planning Your AI Journey**

**Phase 1: Months 1-3 (Foundation)**

**Goals:**

**AI Tools to Introduce:**

| **Tool** | **Purpose** | **Team** | **Timeline** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Training Plan:**

**Success Metrics:**

**Budget:** € \_\_\_\_\_\_\_

**Phase 2: Months 4-6 (Expansion)**

**Goals:**

**Process Changes:**

**Additional Tools:**

**Success Metrics:**

**Budget:** € \_\_\_\_\_\_\_

**Phase 3: Months 7-12 (Optimization)**

**Goals:**

**Advanced Capabilities:**

**Team Development:**

**Success Metrics:**

**Budget:** € \_\_\_\_\_\_\_

**Phase 4: Months 13-18 (Innovation)**

**Goals:**

**Custom Solutions:**

**Competitive Advantages:**

**Success Metrics:**

**Budget:** € \_\_\_\_\_\_\_

**Roadmap Review Schedule**

**Monthly Reviews:**

* What: Progress check
* Who: Team leads
* Duration: 30 minutes

**Quarterly Reviews:**

* What: Strategic assessment
* Who: Management team
* Duration: 2 hours

**Annual Review:**

* What: Complete roadmap revision
* Who: All stakeholders
* Duration: Half day

**Roadmap Risks**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**Implementation Planning**

**Turning Strategy into Action**

**Week 1 Actions**

☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Responsible:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Month 1 Actions**

☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Responsible:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Quarter 1 Actions**

☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
☐ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Responsible:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Communication Plan**

**Who needs to be informed?**

| **Stakeholder** | **Message** | **Method** | **When** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Resource Requirements**

**People:**

**Budget:**

**Tools:**

**Time:**

**Success Criteria**

How will you know you are successful?

**3 Months:**

**6 Months:**

**12 Months:**

**18 Months:**

**Final Workshop Exercise**

**Your Personal Action Plan**

**Your name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Your organization:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Three biggest takeaways from this workshop:**

**Three actions you will take in the next week:**

**Three actions you will take in the next month:**

**Biggest challenge you anticipate:**

**How you will address it:**

**Support you need:**

**Resources and References**

**Recommended Reading**

**Books:**

* (Add your recommendations)

**Articles and Papers:**

* (Add your recommendations)

**Websites:**

* (Add your recommendations)

**Tools and Platforms**

**Code Generation:**

* GitHub Copilot
* Amazon CodeWhisperer
* Tabnine
* Cursor
* Replit Ghostwriter

**Code Analysis:**

* SonarQube
* DeepCode
* Snyk

**Documentation:**

* Mintlify
* Swimm

**General AI Assistants:**

* ChatGPT
* Claude
* Gemini

**Training Resources**

**Online Courses:**

* (Add your recommendations)

**Certifications:**

* (Add your recommendations)

**Communities:**

* (Add your recommendations)

**Appendix: Glossary**

**AI (Artificial Intelligence):** Computer systems that can perform tasks that typically require human intelligence.

**Copilot:** A type of AI tool that assists developers by suggesting code as they write.

**LLM (Large Language Model):** AI models trained on vast amounts of text that can generate human-like text and code.

**Prompt Engineering:** The practice of crafting effective instructions for AI tools to get desired outputs.

**Token:** A unit of text that AI models process (roughly 4 characters in English).

**Context Window:** The amount of text an AI model can consider at once.

**Hallucination:** When an AI generates plausible but incorrect or nonsensical output.

**Fine-tuning:** Training an AI model further on specific data to specialize it.

**API (Application Programming Interface):** A way for different software systems to communicate.

**ROI (Return on Investment):** A measure of the profitability of an investment.

**SLA (Service Level Agreement):** A contract defining expected service levels.

**SSO (Single Sign-On):** Authentication method allowing access to multiple systems with one login.

**MFA (Multi-Factor Authentication):** Security process requiring multiple forms of verification.

**Contact and Support**

**Instructor:**  
Frank Blau  
Email: fjblau@gmail.com  
Phone: +43 663 0660 1372  
LinkedIn: https://www.linkedin.com/in/frank-blau-2ba9131

**Digital Campus Vorarlberg**  
(Add contact information)

**Workshop Dates:**  
January 20-22, 2026

**Notes**

Use this space for additional notes during the workshop.

**End of Workbook**

Thank you for participating in AI for Software Managers. We hope these frameworks and templates help you lead your team successfully through the AI transformation.