

Anthropic团队如何使用Claude Code

Anthropic的内部团队正在通过Claude Code改变他们的工作流程，使开发人员和非技术人员能够处理复杂项目、自动化任务，并弥合以前限制其生产力的技能差距。

通过与我们自己的Claude Code高级用户进行访谈，我们收集了关于不同部门如何利用Claude Code、其对工作的影响以及对其他考虑采用的组织的建议的见解。

目录

- 2 - Claude Code用于数据基础设施
 - 3 - Claude Code用于产品开发
 - 5 - Claude Code用于安全工程
 - 7 - Claude Code用于推理
 - 9 - Claude Code用于数据科学和可视化
 - 11 - Claude Code用于API
 - 13 - Claude Code用于增长营销
 - 15 - Claude Code用于产品设计
 - 17 - Claude Code用于强化学习工程
 - 19 - Claude Code用于法务
 - 21
-

Claude Code用于数据基础设施

数据基础设施团队为公司各个团队组织所有业务数据。他们使用Claude Code来自动化常规数据工程任务、排除复杂的基础设施问题，并为技术和非技术团队成员创建文档化的工作流程，以便独立访问和操作数据。

主要Claude Code使用案例

使用截图进行Kubernetes调试 当Kubernetes集群宕机且无法调度新pod时，团队使用Claude Code来诊断问题。他们将仪表板的截图输入Claude Code，Claude Code逐个菜单地

指导他们浏览Google Cloud的UI，直到找到指示pod IP地址耗尽的警告。然后Claude Code提供了创建新IP池并将其添加到集群的确切命令，绕过了涉及网络专家的需要。

为财务团队提供纯文本工作流程 团队向财务团队成员展示了如何编写描述其数据工作流程的纯文本文件，然后将其加载到Claude Code中以获得完全自动化的执行。没有编码经验的员工可以描述诸如"查询此仪表盘，获取信息，运行这些查询，生成Excel输出"等步骤，Claude Code将执行整个工作流程，包括询问所需的输入（如日期）。

为新员工提供代码库导航 当新的数据科学家加入团队时，他们被指导使用Claude Code来导航他们庞大的代码库。Claude Code读取他们的Claude.md文件（文档），识别特定任务的相关文件，解释数据管道依赖关系，并帮助新人了解哪些上游源输入到仪表板中。这取代了传统的数据目录和可发现性工具。

会话结束时的文档更新 团队要求Claude Code总结已完成的工作会话，并在每个任务结束时建议改进。这创建了一个持续改进循环，其中Claude Code帮助根据实际使用情况完善Claude.md文档和工作流程说明，使后续迭代更加有效。

跨多个实例的并行任务管理 在处理长时间运行的数据任务时，他们在不同存储库中为不同项目打开Claude Code的多个实例。每个实例都保持完整的上下文，因此当他们在几小时或几天后切换回来时，Claude Code确切地记住他们在做什么以及在哪里停下的，实现了真正的并行工作流程管理而不会丢失上下文。

团队影响

在没有专业知识的情况下解决基础设施问题 解决了通常需要引入系统或网络团队成员的Kubernetes集群问题，使用Claude Code来诊断问题并提供确切的修复方案。

加速入职 新的数据分析师和团队成员可以快速理解复杂系统并在没有大量指导的情况下做出有意义的贡献。

增强支持工作流程 可以处理更大的数据量并识别异常（如监控200个仪表盘），这对人类来说是不可能手动审查的。

启用跨团队自助服务 没有编码经验的财务团队现在可以独立执行复杂的数据工作流程。

数据基础设施团队的顶级建议

编写详细的Claude.md文件 您在Claude.md文件中记录工作流程、工具和期望越好，Claude Code的表现就越好。当您有现有模式时，这使Claude Code在设置新数据管道等常规任务中表现出色。

对敏感数据使用MCP服务器而不是CLI 他们建议使用MCP服务器而不是BigQuery CLI来保持对Claude Code可以访问的内容的更好安全控制，特别是对于处理需要记录或有潜在隐私问题的敏感数据。

分享团队使用会话 团队举行了成员向彼此演示其Claude Code工作流程的会话。这有助于传播最佳实践，并展示了他们可能不会自己发现的使用工具的不同方式。

Claude Code用于产品开发

Claude Code团队使用他们自己的产品来构建Claude Code的更新，扩展产品的企业功能和代理循环功能。

主要Claude Code使用案例

使用自动接受模式进行快速原型设计 工程师通过启用"自动接受模式" (shift+tab) 并设置自主循环来使用Claude Code进行快速原型设计，其中Claude编写代码、运行测试并持续迭代。他们给Claude不熟悉的抽象问题，让它自主工作，然后在接管进行最终完善之前审查80%完成的解决方案。团队强调从干净的git状态开始并定期提交检查点，这样如果Claude偏离轨道，他们可以轻松恢复任何错误的更改。

核心功能的同步编码 对于涉及应用程序业务逻辑的更关键功能，团队与Claude Code同步工作，提供具有特定实现说明的详细提示。他们实时监控过程以确保代码质量、样式指南合规性和适当的架构，同时让Claude处理重复的编码工作。

构建Vim模式 他们最成功的异步项目之一是为Claude Code实现Vim键绑定。他们要求Claude构建整个功能（尽管这不是优先事项），大约70%的最终实现来自Claude的自主工作，只需要几次迭代即可完成。

测试生成和错误修复 他们使用Claude Code在实现功能后编写全面的测试，并处理在拉取请求审查中识别的简单错误修复。他们还利用GitHub Actions集成让Claude自动处理拉取请求评论，如格式问题或函数重命名。

代码库探索 在处理不熟悉的代码库（如monorepo或API端）时，团队使用Claude Code快速了解系统的工作原理。他们直接向Claude询问解释和代码引用，而不是等待Slack响应，在上下文切换中节省了大量时间。

团队影响

更快的功能实现 成功实现了复杂功能，如Vim模式，其中70%的代码由Claude自主编写。

提高开发速度 可以快速原型化功能并迭代想法，而不会陷入实现细节。

通过自动化测试提高代码质量 Claude生成全面的测试并处理常规错误修复，在减少手动工作的同时保持高标准。

更好的代码库探索 团队成员可以快速理解monorepo的不熟悉部分，而无需等待同事的回应。

Claude Code团队的顶级建议

创建自给自足的循环 设置Claude通过自动运行构建、测试和lints来验证自己的工作。这允许Claude在您不在时工作，并在您返回时为您提供经过验证的解决方案。

从干净的git状态开始 在让Claude自主工作之前，始终提交您的更改。这样，如果Claude做出您不喜欢的更改，您可以轻松恢复。

使用具体的提示进行关键功能 对于重要的业务逻辑，提供详细的实现说明而不是高级描述。这确保Claude理解您的架构决策和约束。

Claude Code用于安全工程

安全工程团队使用Claude Code来增强他们的安全实践，从代码审查到威胁建模和安全工具开发。

主要Claude Code使用案例

自动化安全代码审查 团队使用Claude Code来执行初始安全代码审查，识别常见的安全漏洞模式，如SQL注入、XSS和身份验证绕过。Claude Code可以快速扫描大型代码库并标记潜在的安全问题，让人类审查员专注于更复杂的架构安全考虑。

威胁建模辅助 在设计新功能时，团队使用Claude Code来帮助创建全面的威胁模型。他们描述系统架构和数据流，Claude Code识别潜在的攻击向量、建议缓解策略，并帮助确保没有遗漏关键的安全考虑。

安全工具开发 团队使用Claude Code来快速原型化和开发内部安全工具，如自定义静态分析规则、安全测试脚本和监控仪表盘。这使他们能够快速响应新出现的威胁和特定的安全需求。

事件响应自动化 在安全事件期间，Claude Code帮助自动化初始响应步骤，如日志分析、影响评估和补救脚本生成。这大大减少了响应时间并确保遵循一致的程序。

团队影响

提高安全审查覆盖率 能够审查更多代码并识别人类审查员可能遗漏的微妙安全问题。

加速威胁建模 将威胁建模过程从几天缩短到几小时，同时提高全面性。

更快的工具开发 可以快速开发和部署安全工具来应对新兴威胁。

改进事件响应 减少安全事件的平均响应时间并提高响应的一致性。

安全工程团队的顶级建议

建立安全知识库 创建包含您组织特定安全模式、威胁和缓解策略的详细文档，以训练Claude Code理解您的安全上下文。

验证所有安全建议 始终让人类安全专家验证Claude Code的安全建议，特别是对于关键系统。

保持更新的威胁情报 定期更新Claude Code的威胁情报知识，以确保它了解最新的攻击技术和防御策略。

Claude Code用于推理

推理团队负责优化Claude模型的性能和效率。他们使用Claude Code来分析模型行为、优化推理管道和开发性能监控工具。

主要Claude Code使用案例

模型性能分析 团队使用Claude Code来分析模型推理日志，识别性能瓶颈和异常行为模式。Claude Code可以处理大量的日志数据并提供关于模型性能趋势的见解。

推理管道优化 Claude Code帮助团队优化推理管道，包括批处理策略、内存管理和GPU利用率。它可以建议配置更改并预测性能改进。

A/B测试分析 在测试不同的模型配置时，团队使用Claude Code来分析A/B测试结果，识别统计显著的差异并提供关于哪些配置表现最佳的建议。

监控仪表盘开发 团队使用Claude Code来开发和维护实时监控仪表盘，跟踪关键性能指标如延迟、吞吐量和错误率。

团队影响

提高模型效率 通过优化推理管道，显著提高了模型的吞吐量和响应时间。

更好的性能监控 实现了对模型性能的实时监控和预警，减少了停机时间。

数据驱动优化 基于详细的性能分析做出优化决策，而不是依赖猜测。

加速实验周期 更快地迭代和测试新的优化策略。

推理团队的顶级建议

建立全面的监控 确保您有详细的性能指标和日志记录，以便Claude Code可以提供有意义的分析。

自动化常规分析 设置Claude Code来自动执行常规性能分析任务，释放团队时间专注于更复杂的优化挑战。

结合领域专业知识 将Claude Code的分析与团队的深度技术知识相结合，以获得最佳结果。

Claude Code用于数据科学和可视化

数据科学团队使用Claude Code来加速数据分析、创建可视化和构建机器学习模型。

主要Claude Code使用案例

探索性数据分析 团队使用Claude Code来快速探索新数据集，生成描述性统计、识别模式和创建初始可视化。Claude Code可以建议相关的分析方法并自动生成代码来执行这些分析。

自动化报告生成 Claude Code帮助团队创建自动化的数据报告，包括图表、表格和解释性文本。这些报告可以定期生成并分发给利益相关者。

机器学习模型开发 在构建ML模型时，团队使用Claude Code来协助特征工程、模型选择和超参数调优。Claude Code可以建议适当的算法并帮助解释模型结果。

数据清理和预处理 Claude Code帮助自动化数据清理任务，如处理缺失值、异常值检测和数据格式标准化。

团队影响

加速数据探索 将初始数据探索时间从几天缩短到几小时。

提高报告质量 生成更一致、更全面的数据报告。

更快的模型开发 加速机器学习模型的开发和迭代过程。

减少重复工作 自动化常见的数据处理任务，让团队专注于更高价值的分析。

数据科学团队的顶级建议

标准化数据格式 建立一致的数据格式和命名约定，以便Claude Code可以更有效地处理您的数据。

创建分析模板 开发可重用的分析模板，Claude Code可以适应不同的数据集和问题。

验证分析结果 始终验证Claude Code生成的分析结果，特别是对于关键业务决策。

Claude Code用于API开发

API团队使用Claude Code来设计、开发和维护Anthropic的API服务。

主要Claude Code使用案例

API设计和文档 团队使用Claude Code来设计RESTful API端点，生成OpenAPI规范，并创建全面的API文档。Claude Code确保API设计遵循最佳实践并保持一致性。

自动化测试生成 Claude Code帮助生成全面的API测试套件，包括单元测试、集成测试和端到端测试。它可以创建测试用例来覆盖各种场景和边缘情况。

性能优化 团队使用Claude Code来分析API性能指标，识别瓶颈并建议优化策略。这包括数据库查询优化、缓存策略和负载均衡配置。

错误处理和监控 Claude Code帮助实现强大的错误处理机制和监控系统，确保API的可靠性和可观察性。

团队影响

更快的API开发 显著减少了从设计到部署的API开发时间。

提高API质量 通过自动化测试和最佳实践，提高了API的可靠性和性能。

更好的文档 生成更全面、更准确的API文档。

增强监控 实现了更好的API性能监控和错误跟踪。

API团队的顶级建议

遵循API设计标准 建立清晰的API设计标准和约定，Claude Code可以帮助执行这些标准。

投资于测试基础设施 建立强大的测试基础设施，以便Claude Code可以有效地生成和运行测试。

监控关键指标 确保您监控关键的API性能指标，以便Claude Code可以提供有意义的优化建议。

Claude Code用于增长营销

增长营销团队使用Claude Code来优化营销活动、分析用户行为和自动化营销流程。

主要Claude Code使用案例

营销活动分析 团队使用Claude Code来分析营销活动的效果，包括转化率、用户获取成本和生命周期价值。Claude Code可以识别表现最佳的渠道和策略。

A/B测试设计和分析 Claude Code帮助设计营销A/B测试，确定样本大小，并分析测试结果以确定统计显著性。这确保了营销决策基于可靠的数据。

用户细分和个性化 团队使用Claude Code来分析用户行为数据，创建详细的用户细分，并开发个性化的营销策略。

内容生成和优化 Claude Code协助生成营销内容，包括电子邮件、广告文案和社交媒体帖子，同时确保品牌一致性和消息传递的有效性。

团队影响

提高营销ROI 通过数据驱动的优化，显著提高了营销活动的投资回报率。

加速实验速度 能够更快地设计、执行和分析营销实验。

更好的用户理解 通过深入的用户行为分析，获得了对目标受众的更深入理解。

自动化营销流程 自动化了许多重复的营销任务，释放团队时间专注于战略工作。

增长营销团队的顶级建议

建立数据管道 确保您有清洁、可访问的营销数据，以便Claude Code可以提供准确的分析。

定义清晰的KPI 建立明确的关键绩效指标，Claude Code可以帮助跟踪和优化这些指标。

保持品牌一致性 在使用Claude Code生成内容时，确保有明确的品牌指南和消息传递框架。

Claude Code用于产品设计

产品设计团队使用Claude Code来加速设计流程、进行用户研究分析和创建设计文档。

主要Claude Code使用案例

用户研究分析 团队使用Claude Code来分析用户访谈、调查数据和可用性测试结果。Claude Code可以识别关键主题、用户痛点和设计机会。

设计系统维护 Claude Code帮助维护和更新设计系统，确保组件的一致性，并生成设计系统文档。

原型和线框图生成 团队使用Claude Code来快速生成原型和线框图的代码，加速从概念到可测试原型的过程。

可访问性审查 Claude Code协助进行可访问性审查，识别潜在的可访问性问题并建议改进方案。

团队影响

加速设计流程 显著减少了从概念到原型的时间。

提高设计质量 通过系统化的分析和审查，提高了设计的质量和一致性。

更好的用户洞察 通过深入的用户研究分析，获得了更深入的用户洞察。

增强可访问性 改善了产品的可访问性和包容性设计。

产品设计团队的顶级建议

建立设计语言 创建清晰的设计语言和原则，Claude Code可以帮助执行这些标准。

结构化用户研究数据 以结构化的方式组织用户研究数据，以便Claude Code可以有效地分析。

迭代和验证 使用Claude Code来快速迭代设计想法，但始终通过用户测试来验证结果。

Claude Code用于强化学习工程

强化学习（RL）工程团队使用Claude Code来开发、训练和优化强化学习模型。

主要Claude Code使用案例

RL算法实现 团队使用Claude Code来实现复杂的强化学习算法，包括策略梯度方法、Q学习变体和演员-评论家方法。

环境设计和模拟 Claude Code帮助设计和实现训练环境，包括奖励函数设计和环境动态建模。

超参数调优 团队使用Claude Code来自动化超参数搜索过程，优化学习率、折扣因子和网络架构等关键参数。

实验跟踪和分析 Claude Code协助跟踪训练实验，分析学习曲线，并比较不同算法和配置的性能。

团队影响

加速算法开发 显著减少了实现和测试新RL算法的时间。

提高实验效率 通过自动化超参数调优和实验跟踪，提高了研究效率。

更好的模型性能 通过系统化的优化过程，获得了更好的模型性能。

增强可重现性 改善了实验的可重现性和结果的可比较性。

强化学习工程团队的顶级建议

标准化实验框架 建立标准化的实验框架和评估指标，Claude Code可以帮助执行这些标准。

投资于计算基础设施 确保您有足够的计算资源来支持大规模的RL实验。

保持实验记录 详细记录所有实验参数和结果，以便Claude Code可以提供有意义的分析。

Claude Code用于法务

法务团队使用Claude Code来协助合同审查、法律研究和合规性检查。

主要Claude Code使用案例

合同审查和分析 团队使用Claude Code来审查合同条款，识别潜在的法律风险，并建议修改建议。Claude Code可以快速分析大量合同并标记需要人工审查的关键条款。

法律研究辅助 Claude Code帮助进行法律研究，搜索相关案例法、法规和先例，为法律论证提供支持。

合规性检查 团队使用Claude Code来检查公司政策和程序是否符合相关法律法规，识别潜在的合规性差距。

文档生成 Claude Code协助生成标准法律文档，如隐私政策、服务条款和合同模板。

团队影响

提高审查效率 显著减少了合同审查和法律文档分析的时间。

增强风险识别 更好地识别潜在的法律风险和合规性问题。

标准化法律流程 通过模板和标准化程序，提高了法律工作的一致性。

改善合规性 更好地维护法律合规性和风险管理。

法务团队的顶级建议

建立法律知识库 创建包含相关法律、法规和公司政策的综合知识库。

人工验证关键决策 对于重要的法律决策，始终要求人工律师验证Claude Code的建议。

保持更新的法律信息 定期更新法律知识库，确保Claude Code了解最新的法律变化。

结论

Anthropic团队在各个部门使用Claude Code的经验表明，这个工具可以显著提高生产力、加速创新并改善工作质量。从数据基础设施到法务，每个团队都找到了独特的方式来利用Claude Code的能力来解决他们特定的挑战。

关键成功因素

- 详细的文档：**所有成功的团队都强调了创建详细的Claude.md文件和 workflows 文档的重要性。
- 迭代改进：**团队持续改进他们的Claude Code使用方式，基于实际经验调整工作流程。
- 人机协作：**最佳结果来自于人类专业知识与Claude Code能力的有效结合。
- 安全和验证：**对于关键任务，团队始终保持人工验证和安全检查。
- 知识分享：**团队之间分享最佳实践和使用案例，加速了整个组织的采用。

对其他组织的建议

- 从小规模试点开始，逐步扩展使用范围
- 投资于培训和文档，确保团队能够有效使用工具
- 建立清晰的安全和合规性指南
- 鼓励实验和创新，同时保持适当的监督
- 定期评估和优化工作流程

通过遵循这些原则和学习Anthropic团队的经验，其他组织可以成功地将Claude Code集成到他们的工作流程中，实现类似的生产力提升和创新加速。