

人間のAIに対する判断移譲: 2年間のパラダイムシフト(2024-2026年)と自律型エージェント時代の統治構造

Gemini 4 Argon

1. 序論:「最終判断は人間が行う」という言説の崩壊と構造的要因

人工知能(AI)の社会実装や企業がバナンスに関する従来の議論において、「作業はAIが行い、最終判断は人間が行う」という命題は、長らく倫理的・実務的な安全弁として機能してきた¹。しかし、2024年から2026年にかけてのわずか2年間にソフトウェア開発の最前線で進行した事態を踏まえると、この命題は現状を記述する枠組みとして著しく解像度を欠くものとなっている²。研究者の西尾泰和氏が2026年10月20日に公開した論考『人間のAIに対する判断移譲』および関連する技術史的考察が浮き彫りにしたのは、人間がAIに対して単なる「作業(Task Execution)」を委ねる段階を超え、「実行を許可する判断(Execution Approval)」、さらには「人間に質問・確認すべきかどうかの判断(Consultation Filtering)」までも段階的に明け渡してきたという観測事実である²。2026年8月の段階に至っては、「何が人間の承認を必要とする事項であるか」というメタレベルの判断自体がAIに委ねられ、それがツールの標準設定(デフォルト)として制度化されるに至っている²。

この急速な判断移譲を駆動した根本的な要因は、人間の怠慢や単なる利便性の追求にとどまらず、AIと人間の間に生じた決定的な「処理・コミュニケーション速度の非対称性」にある²。台湾の初代デジタル発展省大臣であるオードリー・タン(Audrey Tang)氏は、2025年9月に発表した『Plurality & 6-Pack of Care』において、「AIが人間よりも1,000倍、あるいは10,000倍速くなる時代において、人間とAIは根本的な非対称性に直面する。放置すればデフォルトの軌道として、人間が『庭(Garden)』となり、AIが上から人間を管理・剪定する『庭師(Gardener)』になる」と警鐘を鳴らした²。

この速度差がもたらす実務上の構造的ジレンマについて、筑波大学教授の岡瑞起氏は2026年10月4日の国際シンポジウム『リバースアライメント: 多元的社会はAIを受容できるか』において、より具体的なメカニズムを指摘している²。岡氏が指摘する「速さ」とは、単一のAIモデルの推論速度ではなく、背後で自律的に協調・対話する多数のAIエージェント群(Swarm)同士のコミュニケーション速度である⁴。人間がAIから届いた1通のメッセージを読み、周囲と相談して返信を作成しているわずかな時間の間に、背後のエージェント群はすでに何十回も探索の方向性を変えて処理を進めている⁴。この非対称な環境下では、人間の注意のボトルネック化と探索への関与機会の喪失という深刻な板挟み(ジレンマ)が発生する⁴。一方では、AIエージェントが判断の分岐点ごとに毎回人間を呼び出して承認を求めると、人間の認知速度やコンテキストスイッチの遅延がシステム全体の進行を止める最大のボトルネックとなる²。事実、コーディングエージェントの運用において、人間への確認が1回発生するだけで数分間のコンテキストスイッチが生じ、10回の確認で30分以上の待機時間が発生することが計測されている⁸。他方では、人間の注意力を奪わないようAIが人間を減多に呼ばない設計にすると、人間はAIエージェント群による探索の行方や意思決定の軌道に関与し、それを導く機会を完

全に失うことになる²。すなわち、2024年から2026年にかけて起きた「人間からAIへの判断移譲」の歴史とは、この「人間のボトルネック化」という実務上の摩擦を解消するために、開発者とツールベンダーが承認プロセスを次々と自動化・バイパスしていった過程にほかならない²。

2. 判断移譲の2年間史(2024-2026年): 受動的提案から「Auto Mode」の標準化まで

2024年以前から2026年8月に至る約2年半の期間において、コーディングエージェントを中心とするAIツール群は、人間の関与形態を根本から書き換えた²。この歴史の変遷は、移譲された判断の性質に基づき3つのフェーズに区分される²。

2.1 第1期(2024年):提案の受動的受理からローカル環境での「Yolo Mode」誕生へ

2024年以前、GitHub Copilotに代表される開発支援AIは、「AIがコードの断片を提案し、人間がそれを受理(Tabキー押下等)することで初めてエディタ上のファイルが編集される」という受動的な仕組みにとどまっていた²。この構図が転換点を迎えたのが2024年3月12日の「Devin.ai」の発表である²。DevinはAIに専用のシェル、コードエディタ、ブラウザを与え、計画立案から実装、テスト実行までを自律的に完遂する「AIソフトウェアエンジニア」という概念を提示した²。当初のDevinは、AIエージェントが活動するための独立した仮想マシン(サンドボックス)をクラウド上に用意するアーキテクチャを採用していた²。

同時期に台頭したAI統合型コードエディタ「Cursor」やその他のIDE拡張機能は、これとは異なる普及経路を辿った²。これらのツールでは、隔離されたサンドボックス環境を構築する作業がユーザー側の任意タスクに委ねられていたため、環境構築の手間を嫌った多数の開発者が、自身のローカルPCの「生の環境」を直接AIに触らせる運用を選択したのである²。そして2024年末には、Devin的な完全自走の利便性をローカル環境でも享受したいという欲求と、ローカル環境で毎回表示される実行承認プロンプトへの強い煩わしさから、人間の承認を完全に省略してAIを走らせる動きが顕在化した²。

その象徴が、2024年12月17日にリリースされたCursor 0.44のAgent機能における「Yolo Mode」の導入である²。「YOLO(You Only Live Once: 人生は一度きりだから思い切ってやろう)」という俗語に由来するこのモードは、人間による逐一の確認・承認を省き、ターミナルコマンドの自動実行を許可するものであった²。導入当初こそ、サンドボックス化されていない生環境でAIに任意のターミナルコマンドを自動実行させることは、重要ファイルの誤削除や上書きを招く危険な行為として警戒されたものの、一度体験された「承認待ちのない自律実行」の利便性は開発者の行動様式を不可逆的に変化させた²。

2.2 第2期(2025年):ファイル編集・コマンド実行判断の放棄と対話インターフェースの高度化

2025年に入ると、承認プロセスの省略はコマンドライン(CLI)型のエージェントツールにも波及し、人間が保持していた「ファイルを書き換える判断」と「コマンドを実行する判断」が段階的に手放されていった²。

2025年2月24日公開の「Claude Code v0.2.9」には、ツール実行の承認確認をスキップするフラグ `--dangerously-skip-permissions` が実装された(当初はインターネット接続を遮断したDocker環境内での使用に限定されていた)²。続く2025年3月18日の「Claude Code v0.2.47」では、Shift+Tabキーの押下によってファイル編集の自動承認を切り替える機能が導入された²。それまで人間は「AIがどのファイルをどのように編集するか」を事前に差分レビューして承認していたが、この機能以降、AIにプロジェクトフォルダ内のファイルを自由に直接編集させる運用へと大きく舵が切られた²。この背景には、「プロジェクトフォルダ内に限定されていればGitでバージョン管理されているため、万が一AIが重要なファイルを破壊してもロールバック可能であり実害はない。むしろロールバックの単位を小さくするためにAIに細粒度で自動コミットさせればよい」という、事前承認から「事後的な可逆性(ロールバック可能性)の担保」へと安全思想を転換させる現場の合理的判断が存在した²。

さらに2025年4月16日に公開されたOpenAIの「Codex CLI」は、初期READMEにおいて段階的な3つの制御モードを明確に定義した²。具体的には、ファイルの読み取りのみを許可して編集や実行には

承認を要する「Suggestモード」、ファイルの自動編集までを許可する「Auto Editモード」、そしてサンドボックス内でのコマンド自動実行までをすべて許可する「Full Autoモード」の3段階である²。同時にCodex CLIには `--dangerously-auto-approve-everything` フラグが用意され、公開当日に短縮別名として `--yolo` への変更提案がなされた²。そして2025年6月25日にはRust版Codexにおいて、承認確認の省略とサンドボックス隔離の解除を同時に行う

`--dangerously-bypass-approvals-and-sandbox` が追加され、同年8月7日にはその公式エイリアスとして `--yolo` が実装された²。こうして2025年末までに、先進的な開発者の間では「ファイルの書き換え」および「ターミナルコマンドの実行」に関する個別判断を手放すことが常態化した²。

また、2025年後半にはAIから人間への「問いかけの質」にも構造変化が生じた。2025年10月16日の「Claude Code v2.0.21」では、選択式の質問専用ツール(後の AskUserQuestion)が導入された²。これ以前は、AIが特定のアクションを起こす際に人間が「承認(Approve)」か「拒否(Reject)」かの二択を迫られるインターフェースであったが、この機能により、AIが自ら複数の実装方針(例:A案とB案)を策定した上で、「どちらの設計方針がよいか? あるいは自由記述で指示するか?」と人間に高度な意思決定のみを仰ぐ対話が可能となった²。

2.3 第3期(2026年): 質問要否の自己判断、AI承認分類器の実装、およびデフォルト設定化

2026年2月から8月にかけての半年間で、判断移譲は最終段階へと突入した。人間が「AIからの質問に答えること」や「承認ボタンを押すこと」自体をボトルネックとみなし、AI自身に「人間に聞くべきか否か」「操作を許可すべきか否か」を判断させるアーキテクチャが確立されたのである²。

2026年2月17日、エンジニアの逆瀬川氏が公開した「Agentワークフローで人間がボトルネックにならないためのSkill設計(anti-human-bottleneck スキル)」は、実務者コミュニティにおける「相談の移譲」を象徴する事例となった²。Claude CodeのSkills機能は、SKILL.md 内の description に記述された条件に合致した際に自律的にコンテキストへ読み込まれる仕組みを持つ⁸。逆瀬川氏の設計の核心は、トリガー条件をユーザーの発話ではなく、「AIエージェントが人間に質問しようとした、あるいは確認・承認を求めようとした直前の内部状態(BEFORE asking the user any question...)」に設定した点にある⁸。

このスキルが発火すると、AIには「人間に聞かずに全部やれ(git push や本番デプロイの許可を求めるな、次に何をすべきか聞かずにゴールから逆算して進め、正しさの確認を求めず自分で検証せよ)」という行動原則が注入される⁸。AIが人間に頼らず自己検証(Self-Verification)を完結させるため、マルチモーダル機能を用いたブラウザ画面の視覚的確認(claude-in-chrome MCPやPlaywrightによるスクリーンショット解析)、コンソールエラーの読み取り、curl によるAPIレスポンス検証などの手段が与えられた⁸。さらに、PRD(製品要求仕様書)のタスクが完了するまでテスト駆動で永続的に反復実行を管理する外側のループ「Ralph Loop」と、ループ内で確認待ち停止を防ぐ内側の「anti-human-bottleneck スキル」を二重構造で組み合わせることで、完全自律ワークフローが構築された⁸。この設計思想において、人間はもはや常駐の監督者ではなく、SMS認証コードの入力、CAPTCHAの突破、生体認証、物理デバイスの操作といった「AIのツールでは物理的に実行不可能な作業」のみを処理するために呼び出される「高レイテンシ・低帯域の外部ツール」として再定義されている⁸。

こうしたユーザー主導の自律化ハックを受け、プラットフォーム提供企業側も「承認する役割」そのものをAIへ移譲する公式機能を相次いで実装した²。2026年3月24日、AnthropicはClaude CodeのTeamプラン向け研究プレビューとして auto mode を公開した²。これはコマンドや操作の実行前に、

作業担当モデルとは別の「分類器 (Classifier)」が操作の危険性を機械的に判定し、安全と認められる操作を人間の介入なしに自動で通過させる仕組みである²。続いて2026年4月23日にはOpenAIのCodexアプリに自動承認レビュー機能 (Automatic approval reviews) が登場し、4月30日にその技術仕様が公開された²。そこではサンドボックスの境界を越える権限要求が発生した際、作業担当エージェントとは独立した「審査担当エージェント」がその妥当性を審査・承認するマルチエージェント型のガバナンス構造が導入された²。さらに同年5月28日にはClaude Codeにおいて、Claude自身が複数のサブエージェントを動的に起動・制御するJavaScriptスクリプトを生成・実行する Dynamic Workflow が導入され、その性能を最大限に引き出すために auto mode との併用が公式に推奨された²。

そして2026年7月10日にClaude Codeの auto mode が全ユーザー向けに一般提供 (GA) されると、続く8月14日にはPro、Max、Teamの各プランにおいて、新規セッション開始時の既定値 (デフォルト設定) が auto mode へと変更された²。この2026年8月14日の仕様変更は、AI開発史における決定的な分水嶺を意味する。すなわち、「どの操作に人間の承認が必要か」をAI自身が審査・判断する運用が、一部のパワーユーザーが自己責任でフラグ (--yolo 等) を立てて有効化する例外的な機能から、全ユーザーが最初から利用する「業界の標準設定」へと完全に移行したのである¹。

時期	ツール / 出来事	導入された主要機能・設計仕様	人間が手放した (AI へ移譲した) 判断の領域
2024年以前	GitHub Copilot 等	コード補完の提案と人間による手動受理	なし (作業の補助のみ、編集判断は人間が保持) ²
2024年3月12日	Devin.ai / Cursor	自律型開発エージェント (VM隔離) とAIネイティブエディタの登場	計画から実装・テストに至る一連の作業手順の組み立て ²
2024年12月17日	Cursor 0.44	「Yolo Mode」の導入 (生環境でのターミナル自動実行)	ローカル環境におけるターミナルコマンドの個別実行承認 ²
2025年2月24日	Claude Code v0.2.9	--dangerously-skip-permissions フラグの実装 (Docker限定)	隔離環境内でのツール実行に関する事前確認 ²
2025年3月18日	Claude Code v0.2.47	Shift+Tab によるファイル編集自動承認	ファイル書き換え前の事前コード差分レ

		グル (Gitロールバック前提)	ビューと編集承認 ²
2025年4月～8月	Codex CLI (Rust版含む)	3モード定義と --dangerously-bypass-approvals-and-sandbox(--yolo)	サンドボックス境界の維持判断および全コマンドの実行承認 ²
2025年10月16日	Claude Code v2.0.21	選択式質問ツール (AskUserQuestion) の導入	単一行動の諾否判定 (複数案からの選択・方針決定へ移行) ²
2026年2月17日	逆瀬川氏の Skill 設計	anti-human-bottleneck と自己検証 (MCP/Playwright)・Ralph Loop	「人間に質問・確認すべきか否か」の相談判断および検証判断 ²
2026年3月24日	Claude Code	auto mode (研究プレビュー): 分類器モデルによる危険性判定と自動通過	安全性の審査と実行を許可する「承認者 (Approver)」の役割 [cite: 2]
2026年4月23日	OpenAI Codexアプリ	Automatic approval reviews: 別エージェントによる境界越え要求の審査	サンドボックス境界を越える特権操作の一次審査判断 ²
2026年5月28日	Claude Code	Dynamic Workflow: JSスクリプトによるマルチエージェント動的制御	複数エージェントの編成・タスク配分と実行フローの制御 ²
2026年8月14日	Claude Code	Pro / Max / Teamプランにおける新規セッションの auto mode デフォルト化	「何が人間の承認を必要とするか」というメタ判断の常態的移譲 [cite: 2]

3. 判断移譲の理論的・認知的メカニズム: 自動化レベルと認

識論的境界

2024年から2026年にかけて現場で進行した判断移譲は、認知システム工学(Cognitive Systems Engineering)およびヒューマンファクターズ研究の理論的枠組みと照らし合わせることで、その構造的リスクと認識論的な意味合いをより精緻に解剖することができる⁹。

3.1 古典的自動化理論(LOA)の再解釈と「レベル9」の実体化

人間と自動化システムの権限配分に関する最も古典的な基準は、1978年にトーマス・シェリダン(Thomas B. Sheridan)とウィリアム・バープランク(William L. Verplank)が海底遠隔操作システムの研究において提唱した「10段階の自動化レベル(Levels of Automation: LOA)」である⁹。さらに2000年には、ラジャ・パラスラマン(Raja Parasuraman)、シェリダン、クリストファー・ウィッケンズ(Christopher D. Wickens)らがこれを拡張し、情報処理を「情報収集(Information Acquisition)」「情報分析(Information Analysis)」「意思決定と行動選択(Decision and Action Selection)」「行動実装(Action Implementation)」の4段階に分割した上で、各段階に独立した自動化レベルを設定する2軸モデルを確立した⁹。

シェリダンらの10段階スケールにおいて、コンピュータがすべての決定・行動を人間に委ねる手動状態(レベル1)から選択肢の提示・絞り込み(レベル2~3)を経て、中盤以降のレベルは人間の関与の度合いによって明確に階層化されている¹¹。すなわち、レベル4ではコンピュータが1つの選択肢を提案して人間が決定を下し、レベル5では人間が承認した場合に限りコンピュータがその提案を実行する(Human-in-the-Loop: HITL)¹²。続くレベル6では自動実行される前に人間に一定時間の拒否権(Veto)行使猶予を与え(Human-on-the-Loop: HOTL)、レベル7ではコンピュータが自動的に実行した後に必ず人間へ事後報告を行い、レベル8では人間から尋ねられた場合にのみ事後報告を行う¹¹。さらに最上位領域であるレベル9では「人間へ知らせるかどうかをコンピュータ自身が判断し、必要と判断した場合にのみ報告・相談する」形態となり、最終的なレベル10においてコンピュータがすべてを決定して人間を無視し自律行動する(Human-out-of-the-Loop: HOOTL)¹¹。

この分類に2024~2026年の歴史をマッピングすると、2024年以前のGitHub Copilotは「レベル4~5」に位置していたが、2024年末から2025年のYolo Modelによって一気に「レベル7~8」への跳躍が起きたことがわかる²。そして最も注目すべきは、2026年2月の anti-human-bottleneck スキルや2026年3月~8月の auto mode(AI分類器による承認判定)が実現した状態が、シェリダンが1978年に理論化した「レベル9(コンピュータ自身が必要と判断した場合にのみ人間に通知・確認する)」のソフトウェア工学における完全な実体化であるという点である²。さらに、パラスラマンらの4段階モデルに照らせば、従来の自動化システムでは「情報収集」と「情報分析」の自動化レベルを高く設定し、「意思決定」と「行動実装(特に不可逆な操作)」は低く抑えるのが定石であった⁹。しかし現代のAIエージェントは、ツール呼び出し機能(Tool Calling)とAI承認分類器を獲得したことで、情報収集から行動実装に至る4つのすべてのステージにおいて同時にレベル7~9の自律性を獲得するに至っている²。

3.2 委譲の6段階モデルと「認識論的境界(Epistemic Boundary)」

汎用言語モデル(LLM)を用いたエージェントシステムにおいて、「システムがどれほど自律的か」という従来のLOA尺度だけでは、人間側の統治能力の喪失を正確に測ることはできない¹⁰。ダーク・ハームズ=メルビッツ(Dirk Harms-Merbitz)が提唱した「委譲の6段階分類(Six Levels of Delegation)」は、システムの自律性ではなく、「人間のオペレーターの作業に対する理解(メンタルモデル)が最後に更新(Refresh)される場所はどこか」という認識論的な不変量(Invariant)に着目して

いる¹⁰。このモデルでは、各段階において人間が手元に残す唯一の決定（不変量）が定義されており、特に「レベル3」と「レベル4」の間に、従来の自動化スケールが見落としてきた決定的な「認識論的境界（Epistemic Boundary）」が存在することが明らかにされている¹⁰。

委譲レベル	段階の名称	委譲される要素	人間に残る不変量（最後の理解更新点）	人間が読む対象（認識の性質）	対応するAI開発の運用形態
Level 1	One-shot	構文（Syntax）	すべてのアクション（Every action）	作業そのもの（自ら行動し表現のみ取得）	単発の構文照会・スニペット生成 ¹⁰
Level 2	Collaborator	思考・アプローチ（Thinking）	すべての方向性（Every direction）	作業そのもの（提案された複数案の比較）	対話的な設計相談・AskUserQuestion [cite: 2, 10]
Level 3	Reviewer	記述・実装（Writing）	変更の採否（Artefact review）	作業成果物そのもの（コード差分の精読）	2024年以前のCopilot / 手動差分承認 ²
（境界線）	認識論的境界	直接知覚の喪失	成果物の精読から「計測指標の閲覧」への質的転換	事前定義された測定値によって知識が制限される	2025年3月のファイル編集自動承認以降 [cite: 2, 10]
Level 4	Tester / Measurer	実装プロセスの確認	テスト結果・要約統計量（Summary statistic）	作業の測定値（テスト成否・ビルド結果）	Shift+Tab 自動編集＋CI/CDテスト検証 ²
Level 5	Supervisor	個別タスクの検証	許可リストと停止条件（Allowlist & Stopping）	ログと例外通知（AI分類器が捕捉した事象）	2026年8月標準化のAuto mode・審査エー

			condition)		ジェント ²
Level 6	Farmer (農夫)	タスクの選択・順序・時機	環境の設計 (The Environment)	最終的な収穫物 (Harvest: 完成したコード群や計画)	Dynamic Workflow・Ralph Loopによる完全自走 ²

レベル3 (Reviewer) までの段階では、人間はAIが生成した「作業成果物そのもの (ソースコードの差分や文章)」を直接目で読んでいる¹⁰。そのため、もし問題を見落としたとしても、それは単なる「注意力の欠如 (Lapse of attention)」であり、より注意深く読み直すことで回復が可能である¹⁰。しかし、2025年以降のバグコーディングや自動承認モードによって人間がレベル4 (テスト結果の閲覧) やレベル5 (AI分類器のログと停止条件による管理) へ移行すると、人間が知り得る事実「誰かがあらかじめ測定しようと思いついた指標 (テスト項目や分類器のルール)」によって原理的に制限 (Bounded) されることになる¹⁰。レベル5において金曜日の午前3時に重大な障害が発生した場合、その事態を統治している決定は、障害発生時ではなく「火曜日に許可リスト (Allowlist) と停止条件 (Stopping condition) を書いた時点」ですでに下されているのである¹⁰。さらにレベル6 (Farmer: 農夫) に至ると、人間はアウトライン、許可ツール群、スケジュールという「土壌 (環境)」に種を蒔き、後から完成したコードベースという「収穫物 (Harvest)」を受け取るのみとなり、途中のタスク順序やタイミングには一切関与しない¹⁰。

この構造変化は、大規模な実証データによっても裏付けられている。AI支援開発における29,585件のプルリクエスト (PR) をプロセスマイニング手法で解析した研究では、業務の開始主体 (Initiator: 人間かAIか) と最終承認主体 (Approver: 人間かAIか) の掛け合わせによる6つのシナリオ (S1~S6) が分類されている¹⁸。この調査により、AIアシスタントを用いた開発パラダイムにおいては、PRが正式なレビューフェーズを経由することなく初期開発から直接マージへと直行する「サイレント・マージ (Silent Merges)」が構造的なワークフローパターンとして頻発していることが実証されている¹⁸。

3.3 認知的オフローディングとアウトソーシングの分岐、および「自動化の皮肉」

なぜ人間はこれほど急速に認識論的境界を越えて判断を手放すのか。その背景には、人間の認知資源の限界と「自動化バイアス (Automation Bias)」が存在する¹⁵。

まず、AIに毎回の確認を求める運用 (HITL) がいかに機能不全に陥っていたかを示す決定的なデータがある。Anthropicが2026年に公表したデータによれば、Claude Codeにおいて従来の権限確認プロンプトが表示された際、ユーザーはじつに93%のプロンプトを承認していた¹⁵。承認要求が頻繁に発生すると、人間の確認作業は2秒未満で機械的にボタンを押すだけの「ゴム印 (Rubber stamp: 形骸化した自動承認)」へと劣化する³。すなわち、「最終判断は人間が行っている」という建前の裏で、実態としては93%の確率で思考を伴わない承認ボタンの連打が行われていたものであり、これが2026年8月に auto mode がデフォルト化された実務上の根拠となっている²。

しかし、判断の移譲には健全な形態と危険な形態の2種類が存在する。認知科学および知識労働の観点からは、これを「認知的オフローディング (Cognitive Offloading)」と「認知的アウトソーシング (Cognitive Outsourcing)」として厳密に区別する必要がある²⁰。前者の認知的オフローディング (思考資源の再配分) とは、CPUからGPUへ演算処理を逃がしてもシステム設計責任がGPUに移らないのと同様に、低レベルの作業負荷や定型的な確認負荷をAIへ逃がす一方で、問題設定、前提条件

の選択、評価基準の策定、および最終成果物の検収権 (Acceptance Authority) を人間側に保持する使い方である²⁰。浮いた認知資源をより高次のアーキテクチャ設計や批判的検証 (AIに反論や別案を出させる等) に再配分するため、AIの進化に伴って人間の思考の射程が拡張される²⁰。これに対し、後者の認知的アウトソーシング (判断責任の放棄) とは、利用者自身が中身を評価・判断する能力を持たないまま、AIの出力を完成品として受け取り、「AIがそう言ったから」「AIが書いたから」と自らの判断責任を省略する状態を指す²⁰。この状態では、一見流暢で整っているが論理の芯や設計意図が欠落した成果物が量産され、人間がAIの出力に使われる主客転倒が生じる²⁰。

認知的アウトソーシングが常態化した先に待ち受けているのが、ヒューマンファクターズ研究において古くから知られる「自動化の皮肉 (Ironies of Automation)」と「技能低下 (Deskilling)」である⁹。エンズリー (Mica R. Endsley) とケイバー (David B. Kaber) による1999年の動的制御タスク実験が実証した通り、実行段階の自動化は通常運用時のパフォーマンスを向上させるが、人間が実装プロセスから外れることで状況認識 (Situation Awareness) が低下し、「自動化システムが破綻した異常時における人間の回復能力 (Failure Recovery) が著しく悪化する」というトレードオフが存在する⁹。医療AIや航空管制、半自動運転で繰り返し報告されてきたこの現象が、生成AIエージェントによってホワイトカラーの「認知作業」全般へと拡大している¹⁹。日常的な判断をAIに委ねて人間が経験を積む機会を失った結果、AIの分類器やエージェントが対処できない最も複雑で危機的なエッジケースが発生した瞬間にのみ、判断力が減衰し内部構造 (レベル3以下の詳細) を把握していない人間に突如として「最終判断と責任」が突きつけられるのである¹⁰。

4. 現状のシステム設計論：作業・承認・相談の三層分離と多層防御

「毎回人間が手動で確認する (ゴム印化とボトルネック化)」か、「すべてを無防備に丸投げする (破壊的事故のリスク)」かという極端な二極化を回避するためには、システム設計の解像度を根本から引き上げる必要がある³。現在、先進的なAIシステムアーキテクチャにおいて確立されつつある設計原則は、低速な人間のガバナンス領域と人間より1,000倍から10,000倍高速に動作するAIエージェント群 (Swarm) の領域を「認識論的境界」を挟んで構造的に分離した上で、両者を「事前の目的・基準・境界定義」「不可逆な操作に対するエスカレーションゲートと緊急停止機構 (キルスイッチ)」「事後の状態・成果物監査」という3つの制御チャネルで接続する二速度統治モデルに集約される²。

4.1 「判断 (Judgment)」と「決断 (Decision)」および「基準 (Criteria)」の峻別

西尾氏の論考をめぐる技術者コミュニティの議論において、今後人間に残される役割を定義する極めて重要な概念整理が行われている。それは、「判断」と「決断」の分離、および「判断の基準 (メタ規則)」の保持である³。

第一に、合理的な判断材料や評価関数が揃っている領域における「判断 (Judgment)」は、計算速度と情報網羅性に優れるAIに委譲されるべき領域であり、競争社会における差別化要因としての重要度は急速に低下している²³。これに対し、十分な判断材料が存在しない未知の状況下で、価値観の対立を引き受け、結果に対する法的・金銭的・倫理的責任 (Accountability) と覚悟を持って方向性を定める行為は「決断 (Decision / Resolution)」と呼ばれ、これこそが人間に残される本質的機能である³。

第二に、「個別の判断はAIに任せるが、何をAIに任せるかという『基準 (Criteria)』は人間に残す」という原則である³。この基準は、「影響範囲 (Impact Scope)」「可逆性・不可逆性 (Reversibility vs. Irreversibility)」「機械的検証可能性 (Mechanical Verifiability)」という3つの軸の組み合わせによっ

て操作的に定義される³。すなわち、第一の軸である影響範囲においては、操作がローカル環境や内部の実験ブランチに閉じるか、それとも外部システム、顧客、金銭移動、本番環境へ波及するかが問われる³。第二の軸である可逆性においては、Gitのロールバックや下書きの削除のように後から無コストで元に戻せる操作か、それとも外部へのメール送信、決済の実行、本番データベースの削除のように取り返しがつかない操作かが判別される³。そして第三の軸である機械的検証可能性においては、出力の正しさをテストコード、型検査、ステータスコード、画面キャプチャ解析によって機械的に自動検証できるか否かが評価される⁸。

影響範囲が小さく、可逆的であり、かつ機械的に検証可能な操作は、AIの自律ループ(auto modeと自己検証スキル)に完全に委譲し、記録(ログ)を残しながら高速に進行させる³。反対に、不可逆な操作や目的自体の変更を伴う分岐点にのみ、人間の確認(関所)を実行パス(Action Path)の直前に配置するのである³。

4.2 AI承認分類器とサンドボックス隔離の非代替性

システム設計において最も陥りやすい重大な誤謬は、「AIによる承認分類器(Auto modeや審査エージェント)」と「OSレベルの環境隔離(サンドボックス)」を混同し、一方が他方を代替できると錯覚することである³。両者は全く異なる防護層(Defense-in-Depth)に属しており、明確に分離して設計されなければならない³。

Claude Codeのアーキテクチャを例にとると、制御機構は3つの独立した同心円状の防護レイヤーで構成されている³。最も内側のコアで自律的なエージェント群と自己検証ループが高速回転し、そのすぐ外側を、別の分類器モデル(AI)が実行前に操作の危険性を確率的に推論・審査する「自動承認審査(Auto mode)」と、どのツールやコマンドの実行を許可・拒否するかをルールベースで制御する「権限モード(Permission Modes)」が取り囲む²。そして最も外側の物理的境界として、コマンド実行後にプロセスがアクセスできるファイルシステムやネットワークの到達範囲をOSカーネルや仮想化のレベルで強制的に制限する「サンドボックス環境(Sandbox Environments)」が配置される³。ここで特に注意すべき技術的死角は、「シェルのサンドボックス隔離」と「プロセス全体の隔離」の相違である³。ターミナル上で実行されるシェルコマンドだけをサンドボックス内に閉じ込めたとしても、AIエージェントが直接呼び出すファイル操作ツール、Gitフック、あるいは外部と通信するMCP(Model Context Protocol)サーバー群がホストOSの権限で動作していれば、防いだはずのアクセス経路が裏口として残存してしまう³。

さらに、自律型エージェント特有のセキュリティ脅威として、連鎖的アクションによるリスク増幅、マルチエージェント委譲による権限汚染、およびプロンプトインジェクションによる承認審査の突破という3つのリスクベクトルが存在する¹⁵。第一に、個々のステップを見れば低リスクで無害に見える操作であっても、エージェントが複数ステップを計画・連鎖実行することで、機密データの抽出と外部送信の結合のように全体として高リスクな結果を引き起こす「連鎖的アクションによるリスク増幅(Chained Actions Escalation)」が生じる¹⁵。第二に、オーケストレーター(親エージェント)自体は無害な読み取り専用であっても、データベースの書き込み権限や決済APIを持つサブエージェントへタスクを委譲した場合、システム全体のリスクプロファイルが連鎖の末端にある最も強力な権限を持つエージェントに引きずられる「マルチエージェント委譲による権限汚染(Multi-Agent Delegation)」が発生する¹⁷。第三に、外部のWebページやドキュメントに埋め込まれた悪意あるプロンプトインジェクションにより、AI承認分類器が騙されるだけでなく、人間の承認者に対して提示される確認ペイロード(画面表示)自体が偽装・整形され、人間が危険性に気づかずに承認ボタンを押してしまう脆弱性が指摘されている¹⁵。

したがって、実効性のあるエージェント統治には、モデルの推論（AI分類器）の外側にハードウェア・OSレベルの技術的制限（Hard technical limits）を敷き、リスク階層に応じた厳格な運用マトリクスを適用することが不可欠となる³。

リスク階層	対象となるエージェント操作の具体例	推奨される制御パターン（HITL / HOTL）	承認・隔離・検証の技術的要件
Tier 1: 可逆的な内部操作	ローカルファイルの編集、メールの下書き作成、チケット起票、テスト実行 ²	Human-on-the-Loop (HOTL) / Auto mode による自動実行 ²	Gitによる細粒度自動コミット、Undo（取り消し）機構の担保、事後ログの自動記録 ²
Tier 2: 外部波及・金銭的影響を伴う操作	顧客へのメール送信、返金処理の実行、出張予約の確定、外部APIへのデータ送信 ³	Human-in-the-Loop (HITL) （実行パス内での事前承認ゲート） ¹⁵	実行直前での一時停止、実行される正確なAPIコールとパラメータの明示、金額・宛先の上限設定 ³
Tier 3: 破壊的操作・特権変更	本番データベースのレコード削除、IAM ロール（権限）の変更、本番環境への直接デプロイ ¹⁵	厳格なHITL（二名承認制） [cite: 15]	指名された承認者（Named approver）に加え第二レビューアの承認を必須化、サンドボックス境界の完全分離 ³
Tier 4: 禁止領域（Prohibited）	監査ログの消去・改ざん、無制限のシェル実行と外部通信の同時開放、法的禁止行為 ¹⁵	ハードリミットによる完全遮断 [cite: 15]	モデルや分類器の判断に関わらず、OS・ネットワーク・IAMレベルでツール自体をインベントリから除外 ³

5. 今後の展望とガバナンス課題：逆アラインメントと法規制の衝突（2026–2028年）

ソフトウェア開発の現場で先行した「AIへの判断移譲」は、今後2～3年の間にあらゆる産業領域および物理空間へと波及し、社会制度や国際的な法規制と激しく衝突することが確実視されている¹⁷。

5.1 二つの速度を接続する「リバースアラインメント」とフィジカル領域への波及
 第一の展望は、AIと人間の速度差を調停する新たな協調技術、すなわち岡瑞起氏が提唱する「リバースアラインメント（Reverse Alignment）」の具体化である²。従来の一方向的なアラインメント（人

間がAIを直接制御・調教する枠組み)は、人間側の認知速度がボトルネックとなるため破綻しつつある²。今後の研究開発の中心は、岡氏が提示した「呼び込み時機(Timing)の学習」と「遅延応答(Delayed Response)の影響力の維持」という2つの技術的問いへと移行している⁴。すなわち、背後で高速に動くAIエージェント群(Swarm)が、人間の注意を不必要に奪うことなく、かつ探索の方向性が重要な分岐点に差し掛かった最適なタイミングを自ら学習して人間を介入させられるかという点と、人間が自らの遅いペースで熟考して遅れて返した応答であっても、すでに何歩も先へ進んでしまったAIエージェント群の探索の向きを後から遡及的・効果的に変更できるアーキテクチャを構築できるかという点である⁴。

この問いは、オードリー・タン氏の『Plurality(多元性)』における「垂直アラインメント(AIが頂点から人間を管理する庭師となる構造)」から「水平アラインメント(AIが地域のインフラや『土地の神(Kami)』として、人間のケアの速度に合わせて協調を補助する構造)」への転換とも深く共鳴している²。特に日本社会のように対立回避傾向が強く、明確な合意形成が行われなまま現状が流される「曖昧な停滞(Ambiguous Stagnation)」が起きやすい環境においては、AIへの無自覚な判断移譲がそのまま責任主体の消失につながる危険性が高い²。

また、判断移譲の波はデジタル空間の「バイブコーディング」にとどまらず、ロボット工学や自動化ラボ(自律実験装置)と結びついた「フィジカルAI(Physical AI)」および実験科学の領域へと急速に拡大していく²³。ソフトウェアであればGitによるロールバックが可能であったが、細菌やウイルスの培養実験、化学合成、自律走行、重要インフラの物理制御においては、一度実行された操作は物理的に不可逆である³。ソフトウェア開発で培われた「多少危険でも全部AIに任せる方が楽である」という行動様式がそのままフィジカル領域へ持ち込まれた場合、半自動運転でハンドルから手を離す行為と同様に、忘れた頃に破滅的な物理的アクシデントを引き起こすリスクが極めて高い²³。そのため、最大リスクの事前見積もりと、物理的なキルスイッチ(緊急停止機構)の設計が今後の最重要課題となる¹⁵。

5.2 EU AI法(EU AI Act)等の国際規制と「実質的な人間による監督」の再定義

第二の展望は、現場で進む「承認の自動化(Auto mode)」と、法規制が求める「人間による監督(Human Oversight)」の間の深刻な摩擦と、それに伴うコンプライアンス設計の刷新である¹⁵。

世界で最も包括的なAI規制である「EU AI法(Regulation (EU) 2024/1689)」は、自律型エージェント(Agentic AI)という固有のカテゴリ名を条文中に持たないものの、その「AIシステム」および「汎用AI(GPAI)モデル」の定義によって自律型エージェントを完全に捕捉している¹⁷。特に、外部APIの呼び出し能力(Tool-calling)や自律的な行動範囲の広さは、第51条における「システミックリスクを有するGPAIモデル」の指定判断において重要な評価要素とみなされている²⁵。さらに、人事採用(応募者の自動スクリーニング)、信用評価(与信判断)、医療、重要インフラ管理など、付属書III(Annex III)に規定される領域で自律的に意思決定やアクションを行うエージェントは「高リスクAIシステム(High-risk AI systems)」に分類される¹⁷。

EU AI法の施行スケジュールに関しては、2025年2月2日に禁止AI慣行(第5条)が発効し、2025年8月2日にはGPAIモデルの義務および行動規範(Code of Practice)が発効、さらに2026年8月2日にはAIオフィスによる監督権限および第50条の透明性義務が適用開始となった²⁵。一方、高リスクAIシステムに対する義務(第9条リスク管理、第12条自動ログ記録、第14条人間による監督、第26条導入者義務など)については、2026年6月16日に欧州議会が承認した「AIに関するデジタル・オムニバス規則(Regulation (EU) 2026/1744)」により、当初の2026年8月2日から16ヶ月間延期され、スタンドアロンの高リスクシステムは2027年12月2日(医療機器や車両などの規制対象製品に組み込まれるAI

は2028年8月2日)からの全面適用となった¹⁵。

この法規制下において特に重要な意味を持つのが、第14条の「人間による監督」と第12条の「自動ログ記録」の連動である¹⁵。EU AI法第14条は、高リスクAIシステムに対し、人間がシステムの能力と限界を理解し、「自動化バイアス (Automation Bias) を常に認識した上で」出力を解釈し、それを無視・反転 (Override) させたり、「停止ボタン (Stop button)」によってシステムを即座にシャットダウンできる設計を義務付けている¹⁵。規制当局および専門家の解釈において、エージェントに対する監督とは「人間が後からダッシュボードやログを眺めること」ではなく、不可逆なアクションの実行パス内に承認ゲートを置くことを意味する¹⁵。同時に、CEN/CENELEC JTC 21が策定を進める整合規格ドラフト (prEN 18229-1: ログ記録・透明性・人間による監督、および prEN ISO/IEC 24970: AIシステムログ記録)のもとでは、エージェントが行ったすべてのツール呼び出し、アクセスしたデータ、意思決定の連鎖を自動記録することが求められる²⁵。ここで決定的なのは、第12条のログから「人間による承認率およびオーバーライド(却下・修正)率」が算出可能であるという点である¹⁵。もし監査ログにおいて、Anthropicの事例のような「承認率93%~100%、平均判断時間2秒未満」という記録が残っていれば、規制当局から「実効性のある人間の監督 (Effective human oversight) が機能しておらず、違法なゴム印承認である」と判定されるリスクが生じる¹⁵。

規制適用日 / 条項	対象カテゴリ・関連法規	自律型AIエージェントおよび判断移譲に対する具体的な規制・技術要件
2025年2月2日 (第5条)	禁止AI慣行 (Unacceptable Risk)	サブリミナル操作、脆弱性の悪用、社会的スコアリング等の全面禁止 (違反時は最大3,500万ユーロまたは世界売上高7%の制裁金) ²⁵
2025年8月2日 (第51条等)	汎用AI (GPAI) モデル義務	技術文書の作成・行動規範の遵守。ツール呼び出し能力や自律的行動範囲がシステムリスク指定の評価対象に ²⁵
2026年8月2日 (第50条等)	透明性義務・AIオフィス監督権	対話型エージェントにおけるAIであることの明示義務、生成コンテンツの透かし (2026年12月2日~既存システムにも拡大) ²⁵
2027年12月2日 (第9・12・14・26条)	付属書III 高リスクAIシステム (Digital Omnibus Regulation (EU) 2026/1744	第12条 (自動ログ): 全ツール呼び出し・API実行・オーバーライド率の記録 (prEN

	による16ヶ月延期後)	ISO/IEC 24970) ¹⁵ 第14条(人間による監督): 自動化バイアスの認識、実行パス内の承認ゲート、停止ボタンの実装(prEN 18229-1) ¹⁵
2028年8月2日(付属書I)	規制対象製品組込み高リスクAI	医療機器、産業機械、車両等の物理的製品(フィジカルAI)に組み込まれた自律システムへの高リスク義務全面適用 ²⁷

すなわち、2027年12月の高リスク義務全面適用に向け、企業は「すべての操作に人間の承認プロンプトを出して形骸化させる」ことも、「すべてを auto mode でブラックボックス化させる」ことも許されない¹⁵。リスク階層に応じて人間の承認を真に重要な局面(稀で高リスクな決定)にのみ厳選して配置し、人間が実質的に内容を精査して一定の割合でオーバーライドが行われていることを、継続的な監査証跡(Audit Trail)によって証明するガバナンス体制の構築が急務となっている¹⁵。

6. 結論: 自律型AI時代における人間の新たな統治責任

2024年から2026年にかけての2年間で進行した「人間からAIへの判断移譲」は、単なる開発ツールの機能改善ではなく、人間と機械の認知的分業における歴史的な転換点である²。ファイル編集の承認放棄、ターミナルコマンドの自動実行(Yolo Mode)、質問要否の自己判断スキル、そしてAI分類器による自動承認(Auto mode)の標準化という一連の進化は、「人間の認知速度がAIエージェント群のボトルネックになる」という構造的必然から生まれたものであり、もはや時計の針を「すべての最終判断を人間が手動で行う」時代へと戻すことは不可能である²。

しかし、操作の実行承認や相談の判断をAIへ移譲したからといって、人間が結果に対する法的・社会的責任(Accountability)から免除されるわけではない³。形式的に「誰が最後のボタンを押したか」を問う時代は終わり、今後の技術リーダー、システム設計者、および組織の意思決定者には、4つの高次な統治責任(Architectural Governance)が求められる³。

第一の責任は、目的の設定と「決断」の保持である³。合理的に計算可能な「判断」はAIへオフローディングしつつ、正解データが存在しない不確実性下での価値判断と責任の引き受け(決断)を人間の中核業務として明確に定義しなければならない³。第二の責任は、境界(ガードレール)と多層防御の事前設計である³。AI承認分類器(モデルの推論)を過信せず、OS・プロセスレベルのサンドボックス隔離、最小権限の原則に基づくツールインベントリの制限、予算・時間・送信先の上限(Allowlist)をモデルの外部にハードリミットとして構築することが不可欠である³。第三の責任は、可逆性の担保と「認識論的境界」の管理である³。人間が成果物の直接精読(レベル3)から測定値・ログによる管理(レベル4~5)へ移行していることを自覚し、可逆的な操作には自動ロールバック機構(細粒度コミット等)を組み込み、不可逆かつ外部影響の大きい操作にのみ実行パス内の承認ゲートと緊急停止スイッチ(Kill Switch)を配置する設計が求められる²。そして第四の責任は、検収能力の維持と承認の非形骸化である¹⁵。人間の限られた注意力(確認コスト)を真に重要な分岐点と最終成果物の現実状態の検収に集中させ、93%のゴム印承認に陥らないよう承認率・オーバーライド率を継続的にモニタ

リングし、自動化バイアスと技能低下 (Deskilling) に抗う検証体制を維持し続けなければならない³。「最終判断は人間が行う」という粗い標語を捨て、「どの層の判断を、いかなる技術的境界と検証可能性の根拠をもってAIに委ね、万が一の逸脱をどこで食い止める構造にするか」を具体的に設計・実装することこそが、自律型AI時代において処理速度の圧倒的な向上と人間の統治(コントロール)を両立させる唯一の道である²。

引用文献

1. 2026年10月21日のAIバズニュース | 安武遼太 | 製造業×AI - note,
https://note.com/major_echium2954/n/n0567d20117fe
2. 人間のAIに対する判断移譲 - 西尾泰和の外部脳,
<https://scrapbox.io/nishio/%E4%BA%BA%E9%96%93%E3%81%AE%E3%81%AB%E5%AF%BE%E3%81%99%E3%82%8B%E5%88%A4%E6%96%AD%E7%A7%BB%E8%AD%B2>
3. <https://lab.main.jp/ai/posts/agent-approval-judgment-delegation/>
4. AI間のコミュニケーション速度は早くなったが人間は遅いまま - 西尾,
<https://scrapbox.io/nishio/AI%E3%81%AE%E3%82%B3%E3%83%9F%E3%83%A5%E3%83%8B%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E9%80%9F%E5%BA%A6%E3%81%AF%E6%97%A9%E3%81%8F%E3%81%AA%E3%81%A3%E3%81%9F%E3%81%8C%E4%BA%BA%E9%96%93%E3%81%AF%E9%81%85%E3%81%84%E3%81%BE%E3%81%BE>
5. Audrey Tang - LessWrong, <https://www.lesswrong.com/users/audrey-tang>
6. 6-Pack of Care: A Manifesto — Civic AI, <https://civic.ai/manifesto/>
7. 濱田 太陽 (Hiroaki Hamada) - マイポータル - researchmap,
<https://researchmap.jp/hirotaiyohamada>
8. https://nyosegawa.com/posts/claude-code-verify-command/?utm_source=chatgpt.com
9. Trigpoint · Levels of automation,
<https://trigpoint.pub/collections/hai/levels-of-automation>
10. Six Levels of Delegation - LinuxToaster,
https://linuxtoaster.com/papers/six_levels.pdf
11. a new level of Automation Taxonomy - HFES Europe,
<https://www.hfes-europe.org/wp-content/uploads/2014/06/Save.pdf>
12. 16.422 Human Supervisory Control,
https://ocw.mit.edu/courses/16-422-human-supervisory-control-of-automated-systems-spring-2004/6cb73391272fcbf537e50790c002d186_020504hmancenter.pdf
13. A model for types and levels of human interaction with automation,
https://www.researchgate.net/publication/11596569_A_model_for_types_and_levels_of_human_interaction_with_automation
14. Human in the Loop (HITL) とは？人の判断を残す設計と事例,
<https://opsfield.ai/blog/human-in-the-loop/>
15. Human-in-the-Loop AI Agents: 2026 Guide | NeuralTrust,
<https://neuraltrust.ai/blog/human-in-the-loop-ai-agents>
16. Levels of Automation (From Sheridan & Verplank, 1978),

https://www.researchgate.net/figure/Levels-of-Automation-From-Sheridan-Verplank-1978_tbl1_235181550

17. What the EU AI Act Means for AI Agents | Arthur,
<https://www.arthur.ai/column/what-the-eu-ai-act-means-for-ai-agents>
18. Collaborator or Assistant? How AI Coding Agents Partition Work,
<https://arxiv.org/html/2605.08017v1>
19. AIは人間をバカにするのか？2026年の研究が示す「便利さ」と技能,
<https://note.com/jpss4691/n/n2dd21cb6c2b7>
20. AI利用に関する「オフローディング」と「アウトソーシング」の違い,
<https://zenn.dev/pdfractal/articles/ac905d21d86a9c>
21. リーダーのAI依存症に警鐘 判断力を失う5つのサイン - Forbes JAPAN,
<https://forbesjapan.com/articles/detail/89979>
22. SecurityDays2026Fukuoka_FB-05_小出 | ドクセル,
<https://www.docswell.com/s/hirosk/5R8483-2026-10-16-114502>
23. 人間のAIに対する判断移譲 - 西尾泰和の外部脳 - はてなブックマーク,
<https://b.hatena.ne.jp/entry/s/scrapbox.io/nishio/%E4%BA%BA%E9%96%93%E3%81%AEAI%E3%81%AB%E5%AF%BE%E3%81%99%E3%82%8B%E5%88%A4%E6%96%AD%E7%A7%BB%E8%AD%B2>
24. HITL (ヒューマン・イン・ザ・ループ) とは――AIに任せる業務の,
<https://panhouse.jp/insight/what-is-human-in-the-loop/>
25. EU AI Act & AI agents: How agentic AI is regulated - Deeploy,
<https://deeploy.ai/eu-ai-act-hub/articles/eu-ai-act-ai-agents-agentic-ai-regulated/>
26. AI Agents Under EU Law A Compliance Architecture for AI Providers,
<https://arxiv.org/html/2604.04604v1>
27. EU AI Act: Ultimate Compliance Guide and Checklist,
<https://www.cequence.ai/eu-ai-act>
28. High-level summary of the AI Act | EU Artificial Intelligence Act,
<https://artificialintelligenceact.eu/high-level-summary/>