UNBOUND 2026 RECAP

AIエージェントの作り方を、部品から順に

From Prompt to Product: Build an AI Agent

Madeline Zhang（Senior Software Engineer）／ Innovation Stage

2026年9月16日（水）16:00-16:30 EDT

株式会社100

このセッションの位置づけ

当社が参加した回の中では、もっとも実装寄りの1本でした。エージェントを構成する部品を、システムプロンプト・ツール・メモリ・人の関与・オーケストレーション・評価・可観測性の順に分解し、最後にHubSpotのAgent Builderで1体作る流れを、事前に録画した画面で見せています。本資料は当日の聴講記録と、会場で表示されたスライドのメモからまとめました。

Madeline Zhang

まず、作るべきかどうかを決める

自動化を使う

毎回同じやり方で済むなら、エージェントにしない

- 手順が毎回同じ
- 誤った操作のコストが高い、または取り消せない
- 毎回同じ出力が必要

エージェントを作る

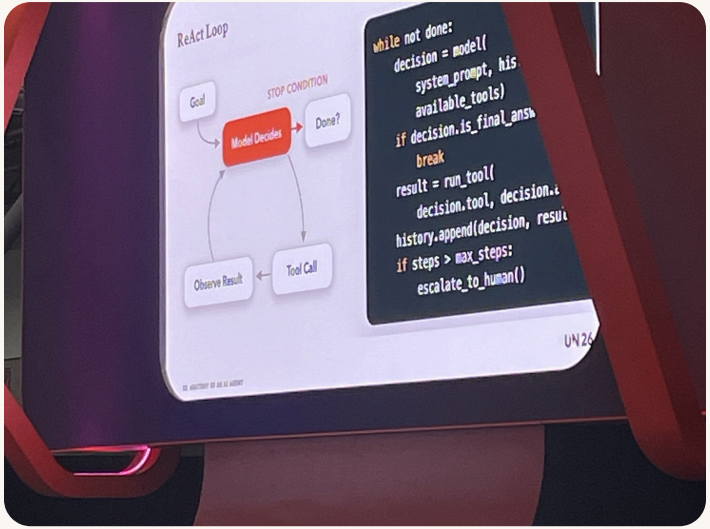
次の5つのうち、3つ以上に当てはまるときだけ作る

- 正しい進め方が1つではなく、いくつもある
- 手順を事前に書き出せない
- 出力が正しいか確かめられる
- 誤りを取り返せる、取り消せる
- 保守の手間に見合う量がある

スライドの見出しは「Build an agent only if 3+」と「Use automation」。例として、1万ドルのワークステーションを買うための調べものが挙げられました。

エージェントが回すループ

- モデルは**行動そのものはしない。次に何をするかを決めるだけ**。実際に動かすのはコード側
- 決める → ツールを呼ぶ → 結果を見る → また決める、を繰り返す
- 止まり方を必ず設計する。理想は目標到達だが、それだけでは止まらない場合がある
- そこで**最大ステップ数**を置く。上限に達したら諦めて人に渡す、という逃げ道を必ず作る



Madeline Zhang

モデルの選び方

スライドの見出しは「Bigger is not better（大きければいいわけではない）」。判断の難しさにモデルを合わせる、という整理でした。

01 判断の難しさで選ぶ

最上位のモデルをすべてに使うのは過剰。メール返信にまで最難関向けのモデルは要らない。

02 推論の深さと速さはトレードオフ

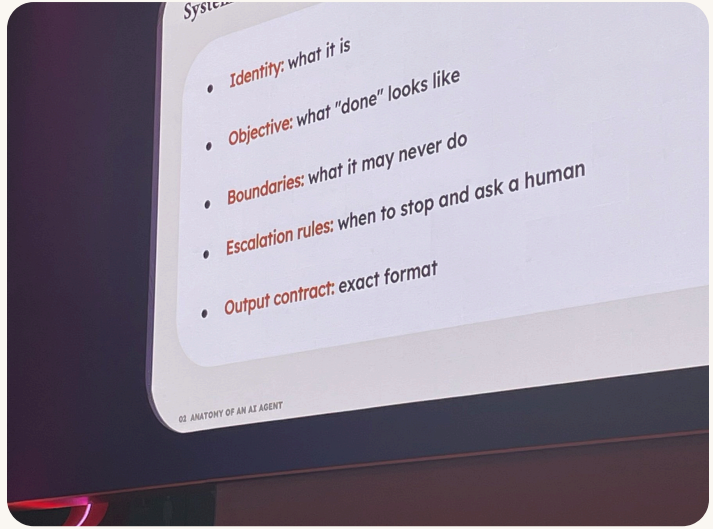
深く考えるモデルは遅い。小型モデルのほうが体感で速い場面は多い。使えるなら推論の強度そのものを調整する。

03 振り分ける

分類のような単純で間違っても傷が浅い処理は安いモデル。判断が要る難所だけ上位モデルに回す。

システムプロンプトに書く5つ

- **Identity**: 何者か。「あなたはシニアのソフトウェアエンジニアだ」
- **Objective**: 終わりの定義。何をもって完了とするか
- **Boundaries**: 禁止事項。「CRMのレコードは変更しない」など
- **Escalation rules**: 人に止まって聞く条件。金額が閾値を超えたら必ず止める
- **Output contract**: 出力の形。検証と観測のために固定する



「あなたは営業担当です、これを売ってください」だけでは毎回違う結果が返ると指摘されました。

Madeline Zhang

ツールの設計で、コストが決まる

ツールは「関数」「スキーマ」「説明文」の3点セット。説明文の質が実行コストに直結します。

- 説明があいまいなまま30個のツールを渡すと、**どれを使うか分からずに片端から試す**。トークンも時間も溶ける
- エージェントは遠回りしてでも正解にたどり着くことがある。動いてしまうので気づきにくい
- エラーメッセージは、人ではなくエージェントに向けて書く。読んで次の手を決める材料になる
- 冪等性を持たせる。二重に呼ばれても顧客に同じメールが何百通も飛ばない作りにしておく

Madeline Zhang

メモリは4層に分けて考える

種類	中身	置き場所
ワーキング	いま処理中のやりとり	コンテキストウィンドウ
会話	そのユーザーとの過去のやりとり	セッションストア
長期・事実	ドキュメント、ポリシー、過去チケット	RAG（ベクトル検索）
構造化	コンタクト、取引、履歴	CRM

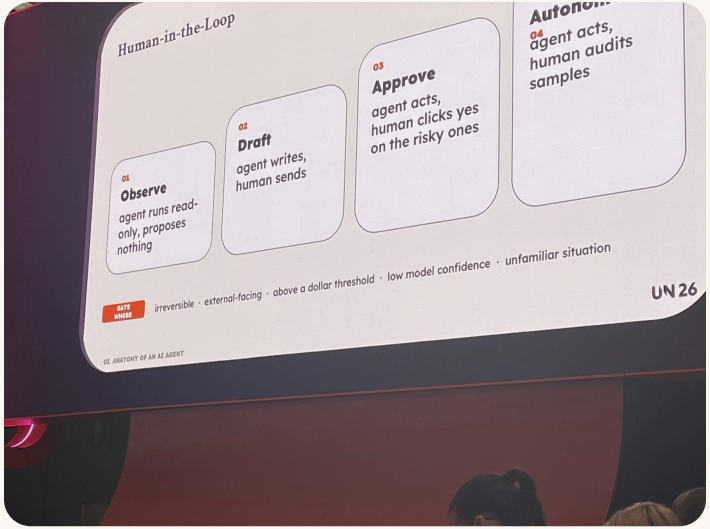
RAGの例として、過去1万件のサポートチケットを埋め込んで、新規チケットに似た上位5件を引く使い方が紹介されました。

コンテキストは、埋まる前から劣化する

- 上限に達していなくても、半分ほど進んだあたりから判断が鈍ることがある
- モデルが劣化したのではなく、前のやりとりが溜まりすぎて混乱している状態だと説明された
- 対処は、上限を待たずに先回りして要約・圧縮し、新しい文脈で始め直すこと

人の関与は4段階で考える

- **Observe**: エージェントは読み取り専用で動き、自分からは何も実行しない
- **Draft**: エージェントが下書きまで作る。送信は人がやる
- **Approve**: エージェントが実行する。ただし危ない操作は人が「はい」を押す
- **Autonomous**: エージェントが実行し、人は後からサンプルを抜き取って監査する



人の確認を挟む条件として、スライドには「取り消せない」「社外に出る」「金額が閾値を超える」「モデルの確信度が低い」「見慣れない状況」が挙がっていました。まずObserveで試し、良ければDraftへ上げる進め方が推奨されました。

Madeline Zhang

複数エージェントの組み方

SINGLE

1体で完結

まずここから。多くの用途はこれで足りる。

ROUTER

振り分け役を置く

依頼の種類を判定して、検索・カスタマーサービス・予約など専門のエージェントへ渡す。文脈を持たない相手に投げない。

MULTI-AGENT

調整役を頂点に置く

各エージェントが自分のループを持ち、上位の調整役が進捗を受け取る。人気はあるが、常に最適とは限らないと釘を刺された。

作った後に必要な2つ

EVALS

エージェント版のテストケース

- 入力と期待出力をそろえた基準セットを用意する
- ツールを足す、プロンプトを直すたびに全部回す
- 出力形式が固定でないものは、別のモデルに判定させる
- 通常のコードと違い、結果が毎回同じにならない前提で組む

OBSERVABILITY

どこで判断を誤ったかを追える状態

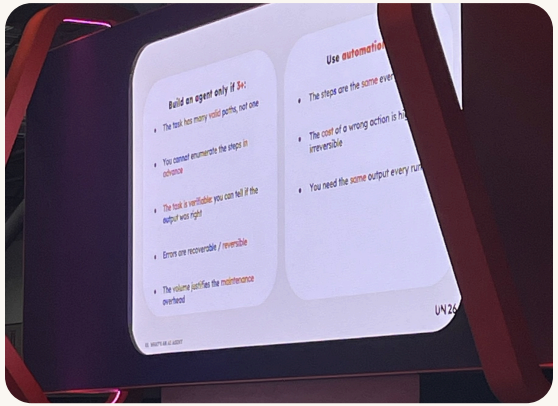
- 呼び出し・プロンプト・結果をすべて記録する
- レイテンシとトークン消費も測る
- 出力が合っているかではなく、最初に誤った判断を探す
- 成功率9割でも、毎日回るワークフローなら失敗は積み上がる

よくある不具合と、直し方

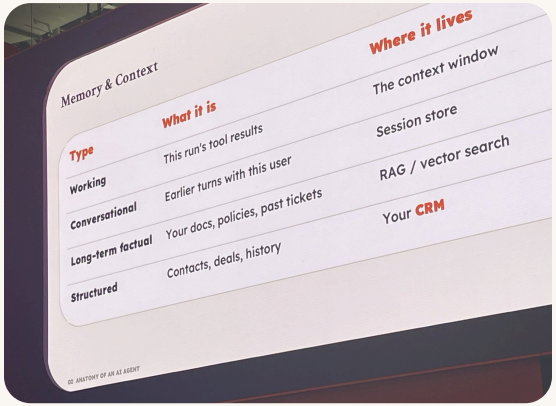
症状	原因	直し方
違うツールを呼ぶ	説明文が曖昧で判別できない	説明文を書き直し、使いどころを明示する
長く動かすと精度が落ちる	コンテキストが膨らんで判断が濁る	劣化を待たず、先回りして要約・圧縮する
事実をでっち上げる	情報が足りないまま答えを出そうとする	検索ツールを与え、「分かりません」と返す許可を明示的に与える

Madeline Zhang

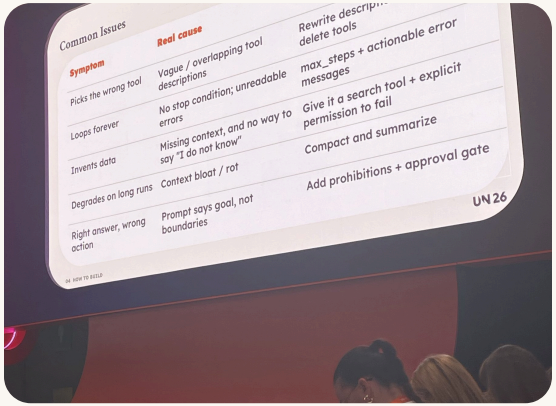
会場で表示されたスライド



エージェントを作るかどうかの判断基準（3ページ）



メモリの分け方（8ページ）



よくある不具合と直し方（13ページ）

締めはAgent Builderのデモ

HubSpotのAgent Builderで1体作る流れを、事前に録画した画面で見せました。

- デモは商談予約エージェント。アクション（ツール）とナレッジ（CRM情報）を持たせている
- 設定項目は、このセッションで説明してきた型と同じ。Identity・エスカレーション・出力スキーマを定義する
- 作成そのものもAIが伴走し、必要そうなツールとナレッジソースを提案してくる
- カレンダー連携はコネクタを利用。翌朝8時の招待を自動で作るところまでを見せた
- 実行のログが残るので、後から何をしたかを追える