

# איך צוות Claude Code משתמש ב-Claude Code

שלושה מהנדסים מהצוות מסבירים איך עברו ממיקרו-ניהול של tool calls להגדרת מטרות, ולמה הם מוחקים פיצ'רים מה-harness כשהמודלים כבר לא צריכים אותם

2026 בספטמבר Thariq Shhipar, Sid Bidasaria, Robert Boyce · Claude · 2

משך: כ-22 דקות · YouTube

סיכום אוטומטי — AI Briefing (aibriefing.dev)

## על המסמך הזה

זו שיחת שולחן עגול קצרה, בערך 22 דקות, בערוץ של Claude. משתתפים בה שלושה אנשים מצוות Claude Code: Thariq Shihpar, שמנחה את השיחה ושואל את רוב השאלות, ו-Sid Bidasaria ו-Robert Boyce, שבונה כרגע את Claude Tag. אין בה שקפים, אין דמו ואין מספרים מסודרים. זו שיחה על הרגלי עבודה, ותועלתה בתיאור מבפנים של הצוות שבונה את הכלי.

הטענה המרכזית: בתוך כשנה עבר הצוות מעבודה ברמת ה-tool call (פרומפט, משוב, אישור הרשאות) לעבודה ברמת המטרה. 70%–80% מהעבודה של אחד הדוברים נעשית היום דרך Claude Tag, agent שחי ב-Slack. לצד זה הם מתארים פילוסופיית מוצר שבה פיצ'רים ב-harness הם פלסטר זמני לחולשות של המודל, ונמחקים כשהמודל משתפר. בדרך הם מספרים איך code review הוביל ל-workflows דינמיים, איך routines יצאו מהלפטופ לענן, ולמה verification הוא התנאי לאמון.



שלושת המשתתפים סביב השולחן. בסרטון אין שקפים, כולו צילומי שיחה

### איך לקרוא את המסמך

- 1. **תקציר מנהלים:** הטענות המרכזיות עם חותמות זמן, ומה לא נאמר.
- 2. **ממשימות למטרות:** איך נראית עבודה יומיומית דרך Claude Tag, ולמה ההקשר של Slack חשוב.
- 3. **טכנולוגיה שמתחלפת כל חודשיים:** AskUserQuestion ו-to-do lists כדוגמאות לפיצ'רים שנבנו ונעלמו.
- 4. **loops ו-routines:** הדרך מהלפטופ דרך dev boxes מרוחקים ועד Claude Code on the Web.
- 5. **מ-code review ל-fan-out workflows:** ביקורת אדברסרית, ו-harness שהמודל כותב לעצמו.
- 6. **Claude Tag בונה את Claude Tag: verification**, ניטור ומשוב בתוך Slack.
- 7. **מה הם מתגעגעים אליו:** מה הולך לאיבוד ומה נפתח כשהעבודה עוברת לרמה גבוהה יותר.
- 8–9. **מה עושים עכשיו ומילון מונחים:** צעדים מעשיים, שאלות פתוחות ומונחים.

**הערת מקור** המסמך הופק אוטומטית מהתמלול ומהפריימים של ההקלטה. התמלול האוטומטי משבש שמות: Claude מופיע בו פעמים רבות כ-Cloud, ו-Claude Tag כ-Cloud Tag או CloudTag, ובמסמך תוקנו השמות לפי תיאור הסרטון. בתמלול לא מצוין מי אומר מה, ולכן ייחוס ציטוט לדובר מסוים מסומן כהסקה מההקשר. הציטוטים מועתקים מילה במילה ונבדקו מול חותמת הזמן שלהם. לא נוסף מידע ממקורות חיצוניים.

# 1. תקציר מנהלים

השיחה מתארת מעבר של צוות Claude Code ממצב שבו המהנדס מאשר כל צעד של המודל, למצב שבו הוא מגדיר מטרה ומקבל תוצאה מאומתת. כל אחד מה-primitives שנבנו בדרך (AskUserQuestion, artifacts, routines, workflows, verification) נשען על קודמיו, וכולם מתחברים בסוף ל-Claude Tag.

- **רוב העבודה כבר לא בטרמינל:** אחד הדוברים מעריך ש-70% עד 80% מהעבודה שלו נעשית ב-Claude Tag. ב-TUI או באפליקציית הדסקטופ הוא משתמש רק כדי ללטש משהו או כשהוא רוצה לנהל את המודל צמוד [0:00].
- **מטרות במקום tool calls:** הצוות הפסיק לעקוב אחרי כל החלטה בתמלול ועבר לתמונה רחבה יותר: יש מטרה, והמודל משיג אותה [1:31].
- **לטכנולוגיה יש תוקף של חודשיים:** פעם התוקף של טכנולוגיה נמדד בשנים, והיום היא משתנה מתחתית כל חודשיים, אולי אפילו מהר יותר [1:31].
- **פיצ'רים ב-harness הם זמניים:** to-do lists נוספו בתקופת Sonnet 3.5 כי המודל נטש משימות ארוכות באמצע, ושנה אחר כך כבר לא היה בהם צורך [3:04].
- **routines בענן הם קפיצת הפרודוקטיביות:** המעבר לקונטיינרים מתארחים היה מסורבל להקמה, אבל אחד הדוברים מעריך שזה הכפיל פי 10 את הפרודוקטיביות שלו [7:45].
- **code review עבר לתמונה הגדולה:** Claude מטפל בהערות הקטנות, והסוקר האנושי מתמקד בגבולות בין שירותים ובמבנה ה-API [9:15].
- **workflows נולדו מ-code review:** fan-out רחב, ביקורת אדברסרית על כל באג מכמה זוויות ואיחוד של התוצאות. Claude יודע לבנות לעצמו את ה-harness הזה [10:46].
- **קוד דטרמיניסטי מגדיל אמון:** כשה-agent כותב לולאת for שמנהלת את ה-sub-agents, ברור שאף פריט לא יידלג [12:20].
- **verification הוא מה שמאפשר לא להסתכל:** PR שמגיע עם טסטים וצילומי מסך מאפשר לוותר על מעקב צמוד אחרי התמלול [16:54].

## מה לא נאמר

- **אין מדידה:** ה-70%–80% וה-10x הן הערכות אישיות, בלי מתודולוגיה ובלי נתון השוואתי.
- **Claude Tag נשאר מעורפל:** לא נאמר אם הוא זמין מחוץ ל-Anthropic, איך מגדירים אותו, מה הוא עולה ואילו הרשאות יש לו ב-Slack.
- **אין סיפורי כישלון:** לא תוארו מקרים שבהם agent אוטונומי שבר משהו, או מתי חזרו לעבודה ידנית.
- **security הוזכר בלי פירוט:** ה-host פתח את הנושא [7:45], אבל הדיון כולו עסק ב-code review.
- **רשימת המחקיקות חסרה:** מלבד to-do lists, לא נאמר אילו פיצ'רים נוספים הוסרו מה-harness. גם auto mode ו-memory הוזכרו רק בחטף.
- **אין נתוני דיוק:** לא נמסר נתון על precision או recall של בוט ה-code review.

## 2. ממשיות למטרות: לעבוד דרך Claude Tag

השיחה נפתחת בהשוואה לשנה הקודמת. אז העבודה עם Claude Code כללה פרומפטים, משוב ואישור של permission prompts. היום, לדברי אחד הדוברים, הבעיות שהוא מעביר ל-Claude Tag מורכבות בהרבה ממשיות כמו לממש מחלקה או לכתוב פונקציה, שבהן Claude Code התחיל.

ההסבר שלו נוגע לסביבה. Claude Tag הוא agent שחי בתוך Slack, ולכן יש לו גישה להקשר המוצרי ולהחלטות שהצוות קיבל לגבי מה המוצר צריך להיות. ההקשר הזה משפר את ההחלטות של ה-agent. דובר אחר מספר שאת שאר העבודה, 20% עד 30%, הוא עושה ב-TUI או באפליקציית הדסקטופ, כשצריך ללטש משהו או כשהוא רוצה לעבוד צמוד למודל.

— **משתתף בפאנל [1:31]** I think we've gone from just caring about the transcript so much and caring about each individual tool call and each individual decision that the model makes to just zoomed out view of, I have a goal and I'd like to achieve this goal

לפי ה-host, הכלים שהצוות בנה לאורך השנה (verification, code review, brainstorming, monitoring) הם primitives נפרדים שבסוף התחברו יחד ל-Claude Tag.

### 3. לבנות על טכנולוגיה שמתחלפת כל חודשיים

לכל מוצר יש תוקף. ההבדל, לדברי הדוברים, הוא שעד היום התוקף של הטכנולוגיה נמדד בשנים, ועם מודלים הוא נמדד בחודשים. מכאן נובע מתח קבוע: צריך לעמוד על החזית ואף לעבור אותה כדי להרגיש איפה הגבול, ובאותו זמן צריך לספק ערך היום, עם המודלים הקיימים. הם מתארים את זה כחצי אמנות וחצי מדע.

— **משתתף בפאנל [1:31]** With models and the AI models, the technology fundamentally shifts underneath you every two months.

הדוגמה המרכזית היא to-do lists. בתקופת Sonnet 3.5 המודל לא החזיק משימות ארוכות: נותנים לו חמישה דברים לעשות, הוא עושה שלושה ומוותר. רשימת משימות מובנית פתרה את הבעיה. שנה אחר כך כבר לא היה בה צורך, כי יש מצבי memory מורכבים יותר שאפשר להישען עליהם. המסקנה שלהם: פיצ'רים ב-harness מכסים על failure modes של המודל הנוכחי. כשהמודל משתפר מוחקים אותם, ובמקומם צריך לבנות כלים חדשים למשימות גדולות יותר.

— **משתתף בפאנל [3:04]** I think that's the perfect example of how you got to be very unattached to the things you're building because they're going to go away fairly quickly.

הדוגמה השנייה היא AskUserQuestion. בהתחלה האינטראקציה עם המשתמש הגיעה רק אחרי שלב התכנון, ואז הוחלט להפוך אותה ל-tool שהמודל יכול לקרוא לו. לדברי מי שבנה אותו, היה קשה מאוד לגרום ל-Claude לקרוא לכלי נכון. אחרי עבודת תכנון ארוכה זה התחיל לעבוד היטב. היום הוא כמעט לא משתמש בכלי: הוא מבקש מ-Claude ליצור artifact ב-HTML, עם דיאגרמות ו-mockups, וה-artifact הוא ששואל אותו את השאלות. לפי ה-host, באותה תקופה צוות קטן בנה proactive agent חדש, ובמקביל נכתבה מחדש מערכת ההרשאות ובנו artifacts-ה.

— **משתתף בפאנל [6:10]** I just create an artifact and the artifact asked me questions.

### 4. routines-ו loops שרצים מרחוק

אחד הדוברים מתאר את הדרך שבה הגיע ל-loops. בהתחלה הכול רץ על הלפטופ, וכשהוא סגר אותו בסוף היום ה-agents נעצרו. מכאן עבר ל-developer boxes מתארחים, שפתרו את התלות בלפטופ אבל חייבו SSH וחזרה לסביבה המרוחקת. השלב הבא היה Claude Code on the Web: קונטיינרים מתארחים שממשיכים לרוץ ברקע.

החיסרון הוא בהקמה. צריך לתת לקונטיינר גישה לסביבת הפיתוח, וזה לא נוח. לדבריו זה שווה את המאמץ.

— **משתתף בפאנל [7:45]** I think it's 10x to my productivity ever since we did that.

כש-Claude רץ בענן אפשר להגדיר routines. הדוגמה שניתנה: פעם ביום לעבור על כל המשוב שהגיע, למיין אותו לפי חשיבות ולתקן את מה שהמודל בטוח ביכולתו לתקן. הדובר מקשר את זה למה שהוא מכנה מסע ה-loops ש-Boris מרבה לדבר עליו (בתמלול מופיע שם פרטי בלבד). הרעיון: לפרוץ את הגבול של session בודד ולכתוב פרומפט לרמה שמעליו, למשהו שמתקן באגים בשבילך.

### 5. מ-code review ל-workflows דינמיים

ה-host מציג את הבעיה: כמות הקוד שנכתב גדלה, ואף אחד לא רוצה שהעבודה שלו תהיה סקירת קוד. לפי אחד הדוברים, השינוי המרכזי הוא ש-Claude עוזר לסוקר האנושי לבחור איפה להשקיע. ב-code review המסורתי הסוקר היה מעלה שלוש הערות קטנות, בעיקר כדי להראות שקרא. את ההערות האלה Claude מטפל בהן היום לבד. מה שנשאר לאדם הוא ההקשר הרחב: למה ה-API בנוי כפי שהוא, ולמה הגבולות בין השירותים עוברים במקום מסוים. ייתכן ש-Claude לא הפנים את הסיבות האלה כשכתב את ה-PR.

— **משתתף בפאנל [9:15]** I think one big thing actually has been Claude really helps human reviewers choose where to focus their time and energy

לדברי ה-code review, host היה הפעם הראשונה שבה הצוות הריץ fan-out גדול: חיפוש רחב של באגים פוטנציאליים ואחר כך איחוד, כלומר שימוש בהרבה test-time compute. (ה-host פונה אליו בשמו, ובתמלול השם מופיע כ-said) מתאר את המבנה. בהתחלה זה היה כמעט harness מותאם אישית: fan-out לאיתור באגים, ואז ביקורת אדברסרית על כל באג משלוש נקודות מבט כדי לבדוק שהוא אמיתי. התוצאה היא רשימה מסוננת של הבאגים שבאמת דורשים תשומת לב.

אותו דפוס מתאים גם לבעיות ביצועים, למחקר כללי ואפילו לתכנון טיול לטאה: עשר שאלות במקביל (מלונות, Airbnb, בתי מרחץ), agents שממיינים, agents שמאמתים, וסיכום בסוף. וכאן החידוש: Claude יודע לתכנן בעצמו את ה-topology של ה-fan-out ולחבר את הפלט של agent אחד לקלט של הבא.

— **Sid Bidasaria (לפי ההקשר) [12:20]** It's like a MapReduce problem. You need to filter back for human consumption

דובר נוסף מצביע על יתרון בצד של האמון. ה-agent כותב קוד שמנהל את ה-sub-agents, ולכן מתקבל שילוב של התנהגות דטרמיניסטית שמוגדרת בקוד עם התנהגות agentic של ה-LLM. לולאת for מבטיחה שכל פריט יקבל את אותו טיפול.

— **משתתף בפאנל [13:50]** I'm like, oh yeah, the for loop, it's not going to skip one of the items. It's going to apply the same technique to all of these things equally

## 6. Claude Tag בונה את Claude Tag: verification ומשוב ב-Slack

Robert Boyce מספר שהצוות משתמש ב-Claude Tag באינטנסיביות כדי לפתח את Claude Tag עצמו. חלק גדול מהעבודה שלו הוא להבטיח שה-agent לא נתקע: סביבת הפיתוח וה-dev loop צריכים להיות פשוטים מספיק כדי ש-Claude יוכל לעשות כל מה שמהנדס אנושי עושה, כולל בדיקה end-to-end. ככל שהתוכנה מורכבת ומשולבת יותר, המשימה הזו קשה יותר.

השינוי העקרוני: הממשק מנותק מהתמלול. ההודעות ב-Slack נשלחות דרך tool. ה-monologue הפנימי של המודל לא מוצג, ויש רק קישור לתמלול המלא. Robert מודה שבהתחלה זה הפחיד, אבל גם שחרר. הוא כבר לא עוקב אחרי אילו tools המודל מפעיל ועם אילו פרמטרים.

— **Robert Boyce (לפי ההקשר) [13:50]** This is the first time we've had the user interface more fully decoupled from the transcript

— **Robert Boyce (לפי ההקשר) [15:22]** And like being forced even to let Claude cook

דובר אחר מתאר מחזור עבודה שלם על כלי פנימי חדש, כולו ב-Slack ונגיש גם מהטלפון. Claude Tag מאתר בעלי עניין שכדאי לדבר איתם, מכין mockups, כותב את המימוש, מוסיף events, והכלי נפרס פנימית. אחרי הפריסה Claude Tag מנטר את השימוש ומתייג אותו כשמישהו נותן משוב. כשהוא ראה שמשתמשים לא מגיעים לסוף ה-funnel, הוא לא הכתיב את הפתרון שחשב עליו. הוא ביקש מ-Claude להציע רעיונות לשיפור ה-funnel, והביא את הרעיון שלו רק כדוגמה.

ה-host מסכם שלושה primitives: verification, code review, וחיבור למקורות משוב (Slack, GitHub, metrics, issues). הוא מדגיש במיוחד את PR verification של Claude ב-Claude Code מגיע עם טסטים וצילומי מסך. בדוגמה שנידונה, Claude התבקש להקליט את עצמו משתמש ב-TUI. הדובר בכל זאת עשה clone ובדק בעצמו, ותיאר את זה כבדיקת שפיות.

— **משתתף בפאנל [16:54]** And so when Claude creates a PR for me for Claude Code, it tests it, it sends me screenshots for what it will look like, and then you get confidence

## 7. מה הם מתגעגעים אליו בהנדסת תוכנה של פעם

ה-host מודה שקשה לעקוב אחרי כל ה-primitives החדשים (auto mode, memory, workflows), ואומר שזה החלק הקשה בעבודת מהנדס תוכנה היום. דובר אחד מתגעגע ל-performance engineering, כלומר לחפירה עמוקה במערכת כדי להאיץ אותה. לדבריו Claude כבר טוב ממנו בזה, אבל הוא עדיין נהנה מהשיפורים. את תשומת הלב שלו הוא מפנה לקיצור הדרך מרעיון לאב-טיפוס ומשם ל-production.

דובר שני, שמתאר את עצמו כמי שאוהב פרטים בעיצוב UI, מספר שבילה פעם יום שלם על שחזור ב-CSS של כפתורי Aqua מ-Mac OS 10.4, עם שכבות של radial gradients. היום הוא לא היה עושה את זה ידנית. הוא נזכר שבגיל שבע או שמונה בנה משחק ב-PowerPoint כי עוד לא ידע לתכנת, ואומר שהיום כל תחום הנדסת התוכנה נראה לו בהישג יד.

— **משתתף בפאנל [18:26]** I think Claude is much better at that than I am now, but I still get to reap the benefits of the actual performance improvements, which is nice

— **משתתף בפאנל [21:31]** I feel like software engineering is the profession of change

## 8. מה עושים עם זה: צעדים ושאלות פתוחות

- **לנסח מטרות, לא צעדים:** במקום להכתיב פתרון, להגדיר את התוצאה הרצויה ולצרף את הרעיון שלכם כדוגמה, כמו במקרה של ה-funnel [16:54].
- **להשקיע ב-dev loop שה-agent יכול להפעיל לבד:** סביבה שבה Claude בונה, מריץ ובודק end-to-end בלי עזרה. זה התנאי לעבודה אוטונומית [13:50].
- **לדרוש verification כחלק מכל PR:** טסטים, צילומי מסך או הקלטה של שימוש. בלי זה אין בסיס לוותר על מעקב אחרי התמלול [16:54].
- **להעביר routines לענן:** מיון יומי של משוב ותיקון רק של מה שהמודל בטוח בו. ההקמה מסורבלת, אבל לדברי הדוברים זה משתלם [7:45].
- **לבנות fan-out עם סינון אדברסרי:** לבעיות חיפוש (באגים, ביצועים, מחקר) ולתת ל-Claude לכתוב את ה-orchestration בקוד [10:46].
- **לבדוק מחדש פיצ'רים ב-harness:** כל scaffolding שנבנה כדי לעקוף חולשה של מודל ישן הוא מועמד למחיקה [3:04].

| שאלה פתוחה                                         | למה זה חשוב                                         |
|----------------------------------------------------|-----------------------------------------------------|
| האם Claude Tag זמין מחוץ ל-Anthropic ובאילו תנאים? | כל המודל של עבודה שמבוססת על מטרות בשיחה נשען עליו. |
| איך מודדים את ה-10x ואת ה-70%–80?                  | ההערכות אישיות, ואי אפשר להסיק מהן על צוותים אחרים. |
| מה קורה כש-agent שרץ ב-routine מתקן משהו לא נכון?  | לא תוארו מנגנוני rollback או סף ביטחון.             |
| איך מאבטחים קונטיינר שקיבל גישה לסביבת הפיתוח?     | security הזכר ולא נדון.                             |
| מה עוד נמחק מה-harness מלבד to-do lists?           | הנתון הזה יעזור להבין לאן Claude Code הולך.         |

## 9. מילון מונחים

| מונח            | משמעות                                                          |
|-----------------|-----------------------------------------------------------------|
| Claude Tag      | agent של Claude שחי בתוך Slack, מקבל מטרות ומדווח בהודעות       |
| harness         | שכבת התוכנה שעוטפת את המודל: tools, הרשאות, זיכרון ותזמור       |
| TUI             | ממשק הטרמינל של Claude Code                                     |
| AskUserQuestion | tool שבאמצעותו המודל שואל את המשתמש שאלה מובנית                 |
| artifact        | פלט עשיר, למשל HTML עם דיאגרמות ו-mockups, שיכול גם לשאול שאלות |
| routine         | משימה מתוזמנת שרצה בענן בלי session פתוח                        |

| מונח              | משמעות                                                          |
|-------------------|-----------------------------------------------------------------|
| loop              | עבודה ברמה שמעל ה-session: פרומפט למשהו שמריץ sessions בעצמו    |
| fan-out           | פיצול משימה להרבה agents במקביל ואיחוד התוצאות                  |
| test-time compute | השקעה של יותר inference בזמן הריצה כדי להגדיל את הביטחון בתשובה |
| workflow          | harness דינמי שהמודל כותב בקוד כדי לתזמר sub-agents             |
| verification      | אימות עצמי של המודל: טסטים, צילומי מסך, הקלטת שימוש             |
| auto mode         | מצב עבודה שבו המודל פועל בלי לבקש אישור לכל פעולה               |