

מה Claude Code יודע לעשות ועוד לא ניסיתם

כ-40 טיפים מהשימוש היומיומי ב-Claude Code ב-58 דקות: הגדרות, עבודה יומית, הרחבות ושימושים מעבר לקוד

2026 ביולי Guilherme "Gui" Ferreira · NDC Conferences · 1

משך: כ-58 דקות · YouTube

סיכום אוטומטי — AI Briefing (aibriefing.dev)

על המסמך הזה

זו הרצאה מכנס NDC AI באוסלו. Guilherme "Gui" Ferreira הוא Software Engineer Educator מפורטוגל, שמעביר סדנאות Claude Code בחברות. ההרצאה מיועדת למי שכבר עובד עם Claude Code, ולא למתחילים. לפי ההגדרה שלו זה פורמט rapid fire: בערך 40 טיפים בשעה, בלי להעמיק באף אחד מהם. הוא מדגיש שאין לו שום קשר ל-Anthropic ושווה לא הרצאת מכירה.

הטענה המרכזית: רוב המשתמשים מגיעים ל-5% עד 10% ממה שהכלי יודע לעשות, ושם הם נעצרים. הסיבה היא שהשיפור בתפוקה כבר כל כך גדול שלא נראה שיש טעם לחפור הלאה. ההרצאה מחולקת לארבעה "דליים": Setup שעושים פעם אחת, הרגלים לעבודה היומית, נקודות ההרחבה של הפלטפורמה, ושימושים שאינם קוד. המטרה שהוא מציב צנועה: אם תאמצו שניים או שלושה טיפים, הוא יחשיב את זה להצלחה.

איך לקרוא את המסמך

- **1. תקציר מנהלים:** הטענות העיקריות עם חותמות זמן, ומה לא נאמר בהרצאה.
- **2. Setup: CLAUDE.md**, רזה, ייבוא קבצים עם @, rules, קבצים מקומיים וגלובליים, permissions.deny, status line-ו.
- **3. העבודה היומית, תכנון ומודלים:** Ctrl+G לעריכת התוכנית, ראיין בעזרת rewind, AskUserQuestion, החלפת מודל, /ultrathink effort-ו.
- **4. העבודה היומית, context ועבודה במקביל:** /context, ה-compact, dumb zone, ניקוי bash, memory, goal/-ו mode, remote control, voice, worktrees.
- **5. Platform Layer:** hooks, skills עם סקריפטים, ה-skills המובנים /simplify, batch/-ו loop, הקובץ mcp.json-ו headless mode.
- **6. מעבר לקוד:** Obsidian, סיכום שבועי, ניתוח לוגים, סידור תיקיית Downloads ו-/insights.
- **7-8. מה עושים עכשיו ומילון מונחים:** צעדים מעשיים, שאלות פתוחות והמונחים באנגלית.

הערת מקור המסמך נוצר אוטומטית מהתמליל של ההקלטה ומהשקופיות שהוצגו בה. ה-ASR שיבש הרבה שמות ומונחים. למשל Claude נשמע כמו Cloud, ו-headless נשמע כמו Adlas, ולכן שמות ומונחים עלולים להופיע משובשים. במקומות שבהם יש סתירה בין השקופית לאודיו ציינו זאת במפורש. הציטוטים מועתקים מילה במילה מהתמליל ובבדקו מול חותמת הזמן שלהם. לא נוסף מידע ממקורות חיצוניים.

1. תקציר מנהלים

ההרצאה לא מציגה ארכיטקטורה או מקרה בוחן. היא קטלוג של הרגלים, ורובם נוגעים בשלושה צירים: חיסכון ב-context ובטוקנים, מניעת דליפת סודות, והפסקת ה-babysitting של ה-agent. לכל טיפ כמעט Ferreira מצרף את הבעיה שהוא פותר אצלו ביום-יום, וזה מה שהופך את הרשימה לשימושית.

- **CLAUDE.md קצר עדיף על CLAUDE.md מפורט:** לדבריו, מחקר של ETH Zurich מצא שהוראות רבות וספציפיות פוגעות בביצועים ומעלות את העלות ביותר מ-20%, ושעדיף CLAUDE.md שאדם מתחזק על פני קובץ שנוצר אוטומטית [7:36]. במקום זה הוא מציע שלוש שאלות סיבון [10:41–7:36].
- **סודות נחסמים בהגדרה ולא בתקווה:** permissions.deny בקובץ claude/settings.json מונע קריאה של קבצי env ו-appsettings והרצה של פקודות כמו dotnet user-secrets. מה ש-Claude קורא נכנס ל-context, ולדבריו "if they go to your context window, they go to Anthropic" [12:12–13:42].
- **מתקנים תוכנית בעריכה, לא בפרומפט חדש:** Ctrl+G פותח את ה-plan כקובץ markdown לעריכה [16:45]. כשהבחירה הייתה שגויה, double Esc או rewind/ מחזירים לנקודה בשיחה, כך שהמימוש הקודם לא נשאר ב-context [21:23].
- **חלוקת עבודה בין מודלים:** מתכננים עם מודל חזק ומבצעים עם מודל זול יותר. effort/ משנה את רמת המאמץ מאותה נקודה ואילך, ו-ultrathink/ מעלה אותה לפרומפט בודד [22:55–21:23].
- **היגיינת context:** מומלץ להריץ context/ בסשן ריק כדי לגלות MCP servers ו-skills שנטענים לשווא. מעל 40–50% מתחילה ה-"dumb zone". כדאי להריץ compact ידנית ולפתוח ששן חדש לעיתים קרובות [29:02–25:57].
- **עבודה במקביל בלי בלבול:** worktrees, color/ ו-rename/ כדי להבדיל בין טרמינלים, goal/ עם תנאי עצירה מדיד, ו-claude agents כלוח בקרה אחד לכל המשימות [41:25–38:20].
- **Hook במקום הוראה ב-CLAUDE.md:** PostToolUse ב-settings.json מריץ פקודה מחוץ ל-context window, כך שלא נצרכים טוקנים ושגיאות נתפסות מוקדם. זה מחליף הוראות כמו "Always run the tests" [42:57].
- **Skills הם יותר מקובץ SKILL.md:** אפשר לשים בתוך skill סקריפט ותיעוד ייחוס במקום להקים MCP server [44:31]. ה-skills המובנים batch/, simplify/ ו-loop/ חוסכים לכתוב skills דומים בעצמכם [47:32–46:01].
- **Claude Code הוא לא רק לקוד:** הוא מתאר ניהול vault של Obsidian, סיכום עבודה לקראת שיחת 1:1, ניתוח לוגים ב-p-claude וסידור של 1,626 פריטים בתיקיית Downloads. ההמלצה האחת שהוא נותן למי שלא יאמץ שום טיפ אחר: להריץ insights [52:10–56:46].

מה לא נאמר

- **המחקר של ETH Zurich מוזכר בלי פרטים:** לא נאמרו שם המחקר, המודלים שנבדקו או גודל המדגם. בשקופית מופיע נתון על ירידה בביצועים (קשה לקרוא אותו בפריים, נראה כ"עד 5%") ועל עלייה של 20%+ בעלות. באודיו נשמע רק נתון העלות.
- **ה-dumb zone היא תחושה ולא מדידה:** הדובר עצמו מודה שאין מחקר שמוכיח את סף ה-40–50% [27:31].
- **אין מספרים על התפוקה שלו:** לא הוצגו מדדים לפני ואחרי, חיסכון בטוקנים או עלויות בפועל של goal/ ו-auto-mode. נאמר רק "Tokens, yes, a lot".
- **אין דיון באבטחה של remote control:** הוא תיאר מנהרה ל-Anthropic ששולטים בה מהטלפון, בלי לדבר על הסיכונים או על מדיניות ארגונית.
- **אין הדגמות חיות:** כל הפיצ'רים הוצגו בצילומי מסך. גם הנתון של 96 slash commands הוא ספירה עצמית שהוא לא אימת מעבר לזה.

2. Setup: מה מגדירים פעם אחת

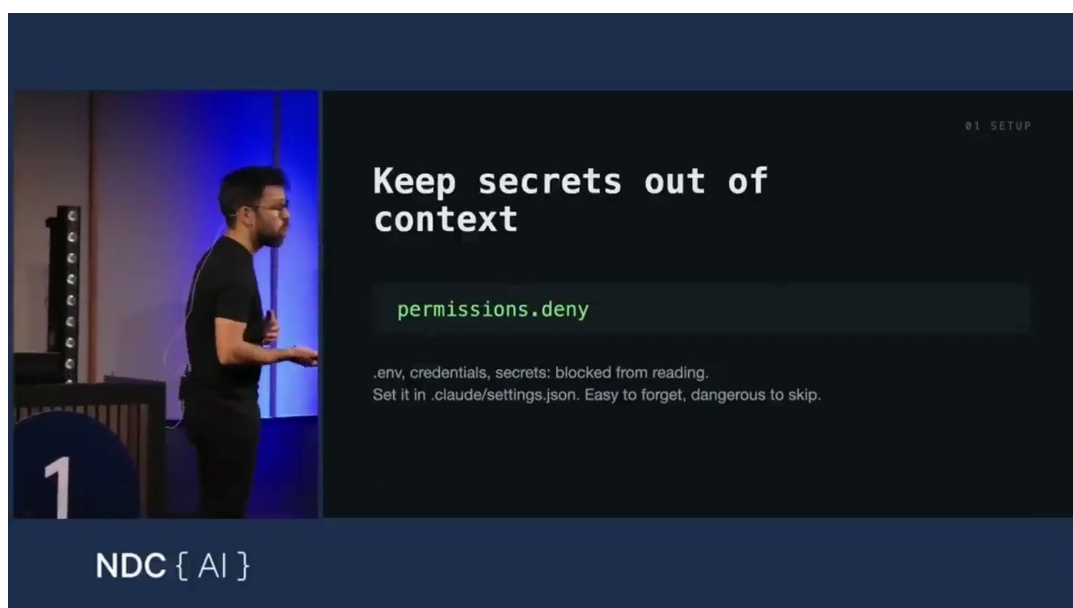


השקופית "Don't add everything to CLAUDE.md": "Agents don't filter. They obey" ונתוני מחקר ETH Zurich על ירידה בביצועים ועלייה של +20% בעלות

Ferreira מתאר את CLAUDE.md כ-"the heart and the soul of the agent" בפרויקט [4:34]. הטיפ הראשון נוגע לכפילויות: מידע שמופיע גם ב-README וגם ב-CLAUDE.md. אפשר לייבא קבצים עם path@, כולל נתיב יחסי וגם קבצים שאינם markdown, למשל schema של API [4:34–6:06].

אחר כך הוא מתאר שינוי בגישה שלו. בעבר היה ממליץ לכתוב ב-CLAUDE.md כל דבר שעולה לראש. המחקר של ETH Zurich השווה בין שלושה מצבים: בלי הקובץ, קובץ שנוצר אוטומטית וקובץ שאדם כתב ומתחזק. לפי המחקר, הוראות רבות וספציפיות מורידות ביצועים ומעלות את העלות "more than 20%" [7:36]. מכאן שלוש שאלות לבדיקת הקובץ: האם המידע רלוונטי ל-95% מהשיחות? האם הוא ספציפי לסוג קבצים מסוים, ואז מקומו ב-skill או בקובץ rules תחת claude. עם glob pattern? והאם ה-agent יכול למצוא אותו לבד? עץ תיקיות, למשל, לא צריך לכתוב, כי הוא מתיישן ו-agent מוצא אותו מהר [9:09–10:41].

העדפות אישיות, כמו TDD שהצוות לא אוהב, נכנסות ל-CLAUDE.local.md שנמצא ב-.gitignore. העדפות שחלות על כל הפרויקטים נכנסות ל-CLAUDE.md הגלובלי בתיקיית claude של פרופיל המשתמש [10:41–12:12].

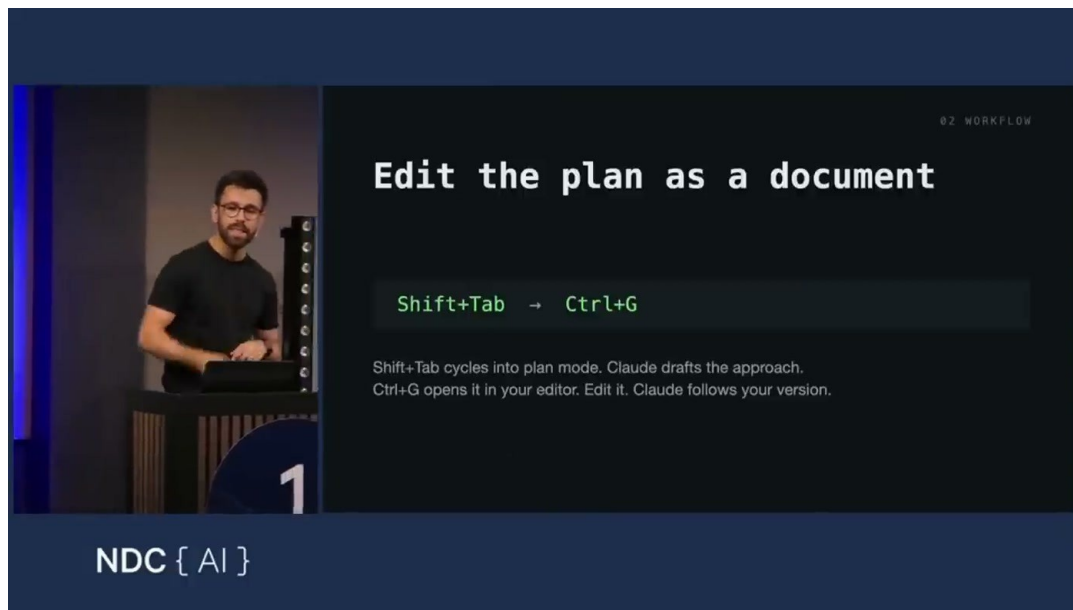


"Keep secrets out of context": קטע permissions.deny ב-.claude/settings.json שחוסם קריאה של קבצי env, credentials וסודות

את הטיפ הזה הוא מבקש במפורש ליישם. כשמבקשים מ-Claude לתקן תקלה בחיבור למסד נתונים, הוא יחפש בקבצי settings, env ו-appsettings. כל מה שהוא קורא נכנס ל-context ונשלח ל-Anthropic. הפתרון הוא settings.json ברמת הפרויקט עם permission deny לקבצים האלה ולפקודות bash מסוימות, כמו dotnet user-secrets. אותו קובץ קיים גם ברמה הגלובלית. בלשונו: "Ideally, no one brings production keys into the local repo, but it happens, you know that, right?" [13:42]

הטיפ האחרון בחלק הזה הוא status line, שאפשר להגדיר בשפה טבעית. הוא מציג git branch, את המודל הנוכחי ואת אחוז ה-context שנוצל. מי שלא יודע מה לשים בו יכול לשאול את Claude אילו שדות זמינים [15:14].

3. העבודה היומית: תכנון, מודלים ורמת מאמץ



"Edit the plan as a document": Shift+Tab עובר ל-plan mode ו-Ctrl+G פותח את התוכנית בעורך

בעבר, כשלא אהב שורה אחת או שתיים בתוכנית, היה כותב פרומפט חדש. היום הוא לוחץ Ctrl+G כשהתוכנית מוצגת. התוכנית נפתחת כקובץ markdown, הוא עורך ושומר, ו-Claude ממשיך מהגרסה הערוכה. בלשונו: "You save time, you save tokens, you save your sanity as well." [18:19]. לדבריו, האינטגרציה מרגישה "like magic".



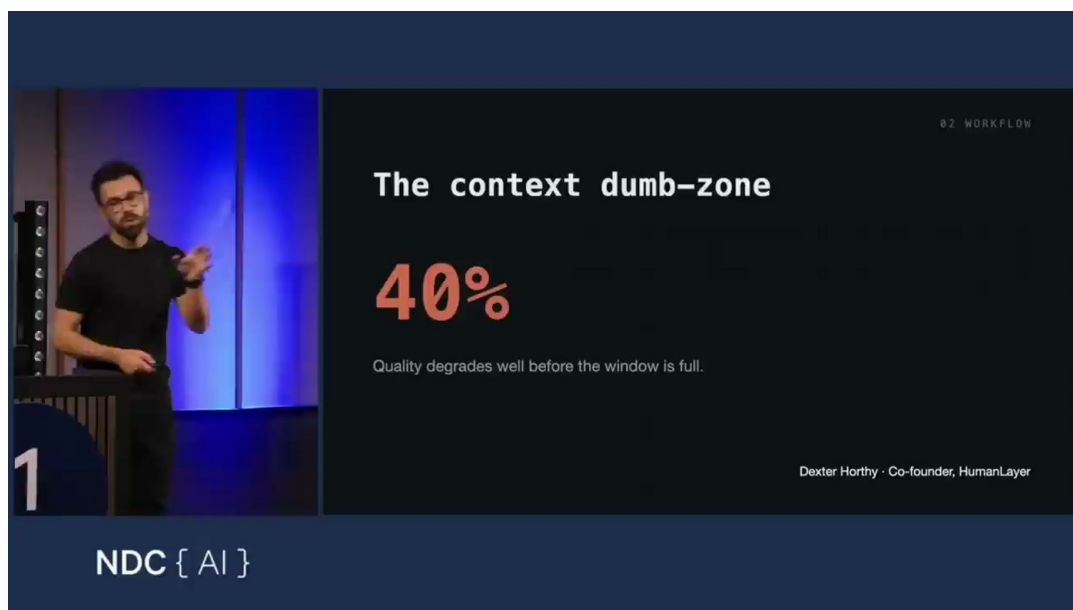
"Let Claude interview you": הפרומפט "interview me about everything I might have missed" שמפעיל את הכל 'AskUserQuestion'

Ferreira עובד ב-specification driven development. הבעיה שהוא מזהה היא ש-Claude ממלא פערים בהנחות שמתגלות רק אחרי המימוש. לכן הוא מכריח אותו להשתמש ב-AskUserQuestion, הכלי שמציג שאלות רב-ברירה ב-plan mode, עם פרומפט בנוסח "please interview me about every single thing that I have inside of this" [18:19]. אחר כך document. That might be an assumption. Don't stop until every single thing is clear." [18:19]. אפשר לעדכן את ה-spec לפי התשובות.

כשמסתבר שהבחירה הייתה שגויה, למשל MVC controllers במקום minimal API, פרומפט תיקון משאיר את כל המימוש הקודם ב-context, והוא ממשיך להשפיע על האיטרציות הבאות. העצה: double Esc או rewind, ו-"[21:23] instead of trying to fix it by moving forward, give one step back to then go forward".

הנקודה השלישית היא כלכלית. התכנון הוא "the heavy lifting" ודורש מודל חזק. ביצוע של תוכנית מפורטת הוא בעיקר מילוי הוראות, ומודל זול יותר יעשה אותו טוב. אפשר להחליף מודל באמצע שיחה. effort/ משנה את רמת המאמץ, למעלה או למטה, לכל המשך השיחה. ultrathink בתוך פרומפט מעלה את המאמץ לפרומפט הזה בלבד, ולדבריו הוא גם נצבע בצבעי קשת [22:55].

4. העבודה היומית: context, שנים ועבודה במקביל



"The context dumb-zone": המספר 40% והשורה "Quality degrades well before the window is full", בייחוס ל-Dexter Horthy, Co-founder של HumanLayer

בזמן ש-Claude עובד, btw/ מאפשר לשאול שאלה צדדית על השיחה בלי לעצור אותו [24:26]. usage/ מציג פילוח לפי מודל ושעות, ולמי שעובד עם API key גם עלות. לדבריו הפילוח שם מפורט יותר מאתגר [24:26–25:57].

האתגר שהוא מציב: לפתוח ששן ריק ולהריץ context/. אם 4% עד 6% מה-context כבר תפוסים לפני שעשיתם משהו, זה בדרך כלל MCP servers ו-skills שהותקנו "just to explore". גם עם חלון של מיליון טוקנים ב-Opus, מעל 40–50% מתחילה מה שהוא מכנה ה-"dumb zone" בשמו של Dexter Horthy מ-HumanLayer (ב-ASR השם נשמע "Dexter Orthy", והשקופית מאשרת Horthy). הוא עצמו מסייג: "We don't have a study that proves this. It's one of those things that might be a feeling, honestly." [27:31]. compact ידנית ולא לחכות לגרסה האוטומטית, ולפתוח שיחה חדשה עם clear/ לעיתים קרובות [27:31–29:02].

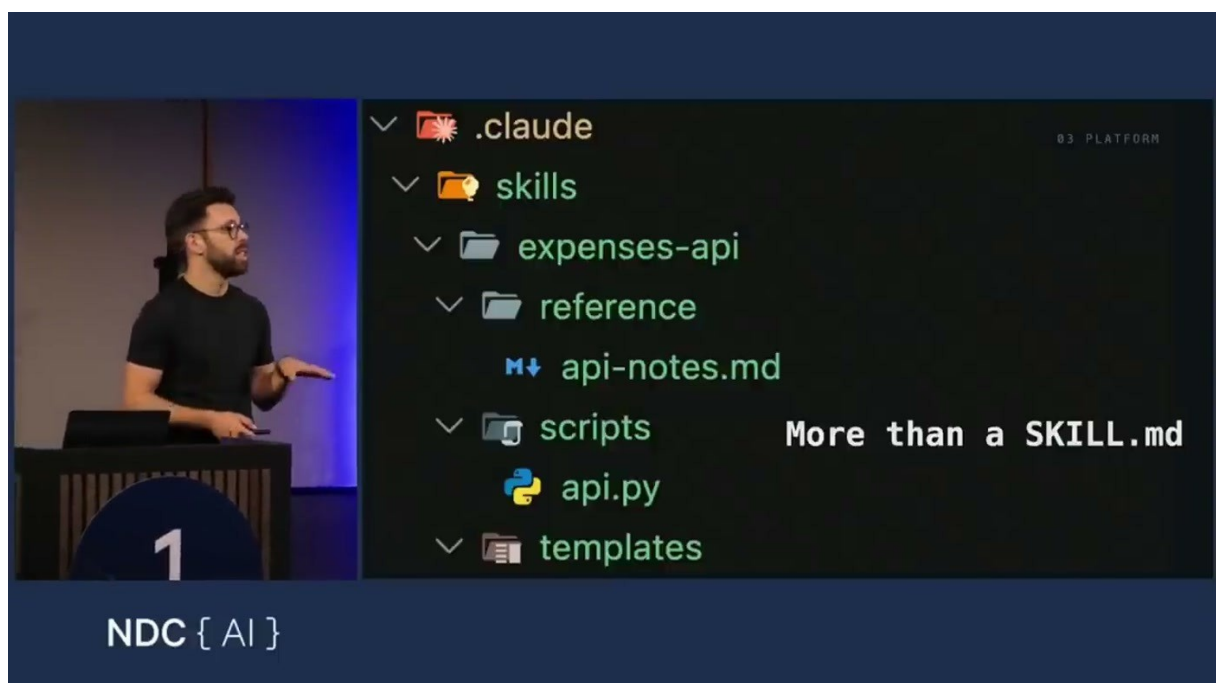
auto-memories נשמרות לתמיד ומתיישנות, ולכן כדאי לבדוק אותן מדי פעם עם memory/. בהקשר הזה הוא סיפר ש-ChatGPT קרא לו פעם [30:33] Emily. סימן קריאה (!) מפעיל bash mode שעוקף את שלב ה-reasoning, אבל הפלט עדיין נכנס ל-[30:33–32:05] context. להמשיך עבודה יש --continue ו--resume. remote-control/ פותח מנהרה ומאפשר להמשיך לאשר צעדים מהטלפון. הדוגמה שלו: להמשיך לעבוד בדרך לאסוף את אביו מבית החולים [33:42]. voice/ מחליף את ה-rubber ducking, ובלשונו [35:13] "the LLM doesn't care if you rumble". אפשר גם להדביק תמונות לטרמינל, ובמק עושים את זה ב-Ctrl+V ולא ב-Cmd+V [36:48].



"Keep Claude working until it's done": הדוגמה "/goal implement the endpoint. don't stop until tests pass/"

לעבודה במקביל על אותו repo הוא משתמש ב-git worktrees. כדי לזהות במבט איזה טרמינל שייך לאיזו משימה, הוא נותן לכל אחד צבע ושם עם color-ו-/[38:20].rename goal/ מגדיר מטרה מדידה עם תנאי עצירה. הדוגמה שלו היא מחולל חשבוניות שלא עוצר עד שיש טסט שמייצר PDF. הוא מוסיף בכנות: "[39:53] Tokens, yes, a lot". לצד זה הוא מזכיר את auto mode, שבו Claude מחליט לבד אילו פעולות בטוחות לאישור. הפיצ'ר חדש ולא מופיע בשקופיות, ולדבריו יש לו עלות נוספת. claude agents פותח dashboard שמראה אילו משימות חסומות, אילו ממתינות לקלט ואילו הסתיימו. הוא מסייג שזה "[41:25] not for everyone".

5. Platform Layer: hooks, MCP-ו skills



"More than a SKILL.md": מבנה התיקיות claude/skills/expenses-api עם תיקיית reference (api-notes.md), scripts (api.py) ו-templates

ה-hooks הם נקודת ההרחבה שלדעתנו הכי פחות מנוצלת. PostToolUse ב-settings.json, אותו קובץ שבו מוגדר ה-deny, מריץ פקודה אחרי שימוש בכלי. בלשונו: "So that command will run outside of the context window, okay? So it doesn't consume your token, so you save money" [42:57]

את התהליך מוקדם כשמהו נשבר. ההמלצה המעשית: כל הוראה ב-CLAUDE.md בנוסח "Always run the tests by the end" צריכה להפוך ל-hook.

לגבי agent skills: ה-description נטען תמיד, ושאר התוכן נטען רק כשצריך. מעבר לזה, skill הוא לא רק "מתכון" בקובץ SKILL.md. אפשר לשים בו סקריפטים ותיעוד ייחוס בנוסח "if this fails, try this". הדוגמה שלו: כשהיה צריך לחבר API עם כמה endpoints שהוא משתמש בהם לעיתים רחוקות, שקל MCP server וויתר. במקום זה ארז סקריפט Python קיים יחד עם README בתוך skill, "like packaging a small application inside of your cloud folder" [44:31].



"Codebase-wide refactoring": הדוגמה `"batch migrate tests/ from Moq to NSubstitute/"`, שמפרקת את העבודה ליחידות ומריצה agent ליחידה ב-worktrees במקביל

לדבריו רוב האנשים מתעלמים מה-skills המובנים ש-Antropic עצמה משתמשת בהם. batch/ מפצל refactoring גדול ליחידות עבודה ומריץ agents במקביל. באודיו הוא אומר "20 to 50" ומיד מתקן ל-"50 to five", ובשקופית כתוב 5-30 יחידות. simplify/ בודק מורכבות ומציע תיקונים, ומחליף את ה-skills ל-code review שאנשים כותבים בעצמם. loop/ מריץ פרומפט בתדירות קבועה, למשל מעקב אחרי CI/CD יחד עם ה-MCP server של GitHub. [46:01-47:32].

Plugins ו-MCP servers מותקנים כברירת מחדל ברמת פרופיל המשתמש, ולכן נטענים לכל פרויקט. זה אומר שבפרויקט שעובד עם Bitbucket ייטענו גם הכלים של GitHub. הפתרון הוא קובץ mcp.json בתיקיית הפרויקט. ה-MCP servers שבו נטענים רק בפרויקט הזה, וכל הצוות עובד עם אותו סט [49:06]. האחרון ברשימה הוא headless mode עם Claude. print-- רץ בלי לחכות לקלט, ואפשר לשלב אותו ב-pipes ובסקריפטים קיימים [50:40] "being the reasoning part of that complex script".

6. מעבר לקוד: Claude Code כעוזר אישי



"One prompt": פרומפט שמבקש לארגן את תיקיית Downloads לפי סוגי קבצים (Images, Documents, Installers, Code, Screenshots) ולהעביר את מה שלא ברור לתיקייה נפרדת (ועוד)

Ferreira ניסה את OneNote, Apple Notes, Notion ו-Obsidian, ואף אחד מהם לא החזיק אצלו. בסוף הבין ש-vault של Obsidian הוא בעצם תיקייה של קבצי markdown מקומיים. היום הוא שואל את Claude Code שאלות על ההערות שלו, מבקש סיכומים ויוצר קישורים בין הערות. את החלק האחרון הוא מכנה "awesome to put in a slide, but they are useless, to be honest" [52:10]. הוא מזכיר ש-Antropic מציעה שלושה מוצרים: Claude Code, Claude Cwork ו-claude.com. לדבריו "One of the problems with the name Cloud Code is that it creates the assumption that is only for code" [52:10].

עוד שימושים שהוא תיאר: סיכום של מה שעשה בשבוע או בחודש האחרון לקראת שיחת 1:1 או הערכה תקופתית, שאפשר לארוז כ-skill. ניתוח לוגים ב-headless mode, ובשקופית מופיעה הפקודה "rank -p | cat *.log | claude". errors by frequency" [53:43–55:15]. וסידור תיקיית Downloads עם 1,626 פריטים לפי סוגים, כך ש-installers וקוד שכבר נמצא ב-GitHub אפשר למחוק מיד [55:15–56:46]. בסיום הוא ממליץ על פקודה אחת: insights/, שמפיקה דוח על דפוסי השימוש שלכם, הצעות לשיפור ה-workflow ורעיונות ל-skills.

Gui Ferreira [56:46] Because I learn more with that command than by reading LinkedIn posts, to —
.be honest

7. מה לעשות מחר בבוקר

- להוסיף **permissions.deny**: ב-claude/settings.json של כל פרויקט צוות וגם ברמה הגלובלית, לקבצי env, appsettings וסודות. זו ההמלצה היחידה שהוא ביקש במפורש ליישם.
- לבדוק את **CLAUDE.md מול שלוש השאלות**: רלוונטיות ל-95% מהשיחות, ספציפיות שמתאימה יותר ל-rules או ל-skill, ומידע שה-agent מוצא לבד. להעביר העדפות אישיות ל-CLAUDE.local.md או לקובץ הגלובלי.
- להריץ **context/ בסטן ריק**: ולהסיר MCP servers ו-skills שלא בשימוש, או להעביר אותם ל-mcp.json ברמת הפרויקט.
- להפוך הוראות "תמיד תריץ טסטים" ל-PostToolUse: hook רץ מחוץ ל-context ותופס שגיאות מוקדם.
- להוסיף את **Ctrl+G**, **rewind** ו-compact **ידני להרגלים**: מתקנים במקום לערום תיקונים ב-context, ומקצרים את השיחה לפני שמגיעים ל-40%–50%.
- לנסות את ה-skills המובנים לפני שכותבים חדשים: **simplify**, **batch** ו-loop.
- להריץ **insights/ פעם אחת**: ולבחור ממנו skill אחד ליצירה.

שאלה פתוחה	למה זה חשוב
מה בדיוק בדק המחקר של ETH Zurich, ועל אילו מודלים?	ההמלצה לקצר את CLAUDE.md נשענת עליו, אבל באודיו ובשקופית הוצגו רק מספרים חלקיים.
האם סף ה-dumb zone של 40%–50 נכון גם בחלון של מיליון טוקנים?	הדובר עצמו אומר שזו תחושה ולא מדידה.
כמה עולים goal/ ו-auto mode בפועל?	נאמר רק שזה "a lot" ושיש עלות נוספת. אין נתונים לתכנון תקציב.
מה פרופיל האבטחה של remote-control/ בסביבה ארגונית?	מדובר במנהרה ל-Anthropic מסשן מקומי, והנושא לא נדון בהרצאה.
לכמה יחידות batch/ מפצל בפועל?	באודיו נאמר 5–50 ובשקופית כתוב 5–30.

8. מילון מונחים

מונח	משמעות
CLAUDE.md / CLAUDE.local.md	קובץ ההוראות של ה-agent בפרויקט. הגרסה המקומית מיועדת להעדפות אישיות ונמצאת ב-gitignore.
permissions.deny	רשימה ב-settings.json שחוסמת קריאה של קבצים או הרצה של פקודות.
status line	שורת מצב מתחת לשורת הפרומפט: branch, מודל ואחוז ה-context.
plan mode	מצב שבו Claude מתכנן לפני ביצוע. נכנסים אליו ב-Shift+Tab.
AskUserQuestion	הכלי שבו Claude שואל את המשתמש שאלות רב-ברירה.
rewind	חזרה לנקודה קודמת בשיחה, ב-double Esc או rewind/.
effort / ultrathink	רמת המאמץ של המודל, לכל השיחה או לפרומפט בודד.
context window / compact	חלון ההקשר שנשלח למודל, וסיכום השיחה כדי לפנות בו מקום.
dumb zone	הטווח שמעל 40%–50 ניצול context, שבו לפי הטענה האיכות יורדת.
auto-memories	זיכרונות ש-Claude שומר לבד בקובץ, ונבדקים עם memory/.
worktrees	כמה תיקיות עבודה של אותו repo, לעבודה במקביל.
goal/	מטרה עם תנאי עצירה מדיד, שעד אליה Claude ממשיך לעבוד.
hook (PostToolUse)	פקודה שרצה אחרי שימוש בכלי, מחוץ ל-context window.
agent skill	חבילת הוראות, סקריפטים ותיעוד שנטענת כשהיא רלוונטית.
plugin	חבילה שמשלבת hooks, skills ו-MCP servers להפצה.
mcp.json.	הגדרת MCP servers ברמת הפרויקט, משותפת לצוות.
headless mode (--print / -p)	הרצת Claude בלי אינטראקציה, לשילוב ב-pipes ובסקריפטים.
vault (Obsidian)	תיקייה של קבצי markdown שמשמשת כמאגר הערות.