

Steve Yegge על Gas Town, שמונה רמות האימוץ של AI ועתיד מקצוע התוכנה

שיחה של כשעה וחצי בפודקאסט The Pragmatic Engineer: מ-Claude Code שמריץ Claude Code, דרך שחיקה ו-value capture, ועד חברות גדולות ש"עוד לא יודעות שהן מתות"

Steve Yegge, Gergely Orosz · The Pragmatic Engineer (פודקאסט; הועלה מחדש ב-X על ידי Ichigo) · תאריך השידור המקורי לא ידוע (הועלה ל-X ב-12.9.2026)

משך: כ-93 דקות · X

סיכום אוטומטי — AI Briefing (aibriefing.dev)

על המסמך הזה

זה פרק של הפודקאסט The Pragmatic Engineer, ולא הרצאה עם שקפים. Gergely Orosz, המגיש (ה-ASR, כותב את שמו "gergie"), משוחח עם Steve Yegge, מהנדס עם כ-40 שנות ותק שעבד ב-Google וב-Amazon, כתב יחד עם Gene Kim את הספר Vibe Coding ובנה את Gas Town, orchestrator בקוד פתוח שמריץ agents. הסרטון הגיע מ-repost ב-X. הכיתוב שם מציג את הדובר כ"Google engineer" ומצטט אותו אומר "Just delete your IDE" ו-"85% of PRs at Google". שום דבר מזה לא מופיע בתמליל. Yegge גם מספר בעצמו שהוא כרגע "unemployed" [1:31].

הטענה המרכזית של Yegge: מודלים כמו Opus 4.5 כבר הפכו את פיתוח התוכנה ל"בעיה הנדסית". מי שעדיין עובד עם Copilot יישאר מאחור. העבודה עוברת מכתובת קוד לניהול של כמה agents במקביל, ומשם ל-orchestration מלא. לצד ההתלהבות הוא מזהיר משני דברים. הראשון הוא "אפקט ערפדי" של שחיקה. השני הוא חלוקת ערך לא הוגנת בין מהנדסים לחברות. לדעתו חברות גדולות יאבדו בערך חצי מהמהנדסים, והחדשנות תעבור לצוותים של עד 30 אנשים.



כרטיס הפתיחה של הפרק: "Steve Yegge, 'famous for his brutally honest rants about the industry and for being right, a lot'. זה הפריים היחיד במסמך. שאר הסרטון הוא צילומי אולפן ופרסומות.

איך לקרוא את המסמך

- 1. **תקציר מנהלים:** הטענות המרכזיות וחותמות הזמן שלהן, ומה לא נאמר.
- 2. **סולם האבסטרקציה:** מקומפילרים ו-assembly ועד הרגע שבו Claude Code שכנע את Yegge.
- 3. **רמות האימוץ וה-IDE:** רמות 4–7, ריבוי Claude Cowork, agents, והבעיה של "אנשים לא יודעים לקרוא".
- 4. **Gas Town: Mayor, Polecats, ו-Crew,** הוויכוח על גודל ה-context, המעבר ל-Dolt ומקרי שימוש אמיתיים.
- 5. **הצד האנושי והארגוני:** אפקט הערפד, value capture, ההשוואה בין Anthropic ל-Google, וחברות גדולות "מתות".
- 6. **קוד שאף אחד לא קורא:** The Bitter Lesson, heresies, תוכנה אישית והחזרה ל-open source.
- 7–8. **מה עושים עכשיו ומילון מונחים:** צעדים מעשיים, טבלת שאלות פתוחות ומילון מונחים.

הערת מקור המסמך הופק אוטומטית מתמליל ההקלטה ומהפריימים שחולצו ממנה. אין בסרטון שקפים, ולכן מופיע בו פריים אחד בלבד. התמליל שהתקבל חלקי, ויש בו קפיצות בין בלוקים (למשל 1:31 ל-4:38, ו-6:09 ל-9:13). שמות ומונחים עלולים להשתבש ב-ASR, ובמקומות האלה זה מצוין. כל ציטוט מועתק מילה במילה מהתמליל ונבדק מול חותמת הזמן שלו. לא נוסף מידע ממקורות חיצוניים.

1. תקציר מנהלים

השיחה נעה בין שלושה צירים. הראשון הוא הטכנולוגיה: עקומות שיפור אקספוננציאליות, orchestration של agents ו-Gas Town. השני הוא האדם: שחיקה, קריאה וזהות מקצועית. השלישי הוא הארגון: פיטורים, value capture וחדשנות שנעלמת מחברות גדולות. Yegge מדבר בביטחון רב ועם מעט נתונים. רוב המספרים שלו הם הערכות או אנלוגיות, ולא מדידות.

- **הסולם עובר מכתובת קוד לניהול agents:** ברמה 4 מסתכלים פחות על diffs, ברמה 5 ה-IDE משמש רק לקריאה, ברמה 6 מריצים כמה agents במקביל, וברמה 7 "I've made a mess" ומתחילים לחשוב על תיאום [23:03].
- **Claude Code שמריץ Claude Code הוא הבעיה הפתוחה:** הניסיונות במהלך השנה נתקעו כי ה-agent Gas Town "would run for a while and it would stop" נבנה כדי לפתור בדיוק את זה [23:03].
- **שני סוגי workers הם פתרון לזינוח על context:** Polecats משימות קצרות ומוגדרות היטב עם context מינימלי. Crew מקבלים context מקסימלי לעבודת תכנון [33:46–32:15].
- **"Gas Town deliberately doesn't work":** לדברי Yegge זה ניסוי, ו-"even opus 4.5 is barely enough". הוא מצפה שחלק גדול מהמורכבות ייעלם כשהמודלים יאומנו לעבודה כזו [33:46].
- **כ-50% מהמהנדסים בחברות גדולות ילכו:** זו אנלוגיה ל"חוגה" שמכוונים בה כמה מהנדסים לפטר כדי לממן tokens לשאר. המספר לא מגובה במקור [18:31].
- **אפקט ערפדי:** פרודוקטיביות גבוהה פי 100 מגיעה עם תשישות ותנומות באמצע היום. נשארת פתוחה השאלה מי לוקח את הערך, החברה או המהנדס [43:30–41:58].
- **החדשנות בחברות גדולות מתה:** ב-Google, לדבריו, היא נעצרה כשיותר אנשים התחרו על פחות עבודה. Anthropic עובדת כמו "hive mind" סביב [48:17–52:56].
- **במודל עצמו כבר לא צריך עוד התקדמות:** "opus 4.5 made this officially an engineering problem". ההמלצה היא חצי שעה ביום עם Claude Code [1:19:25].
- **Heresies:** בקוד שאף אחד לא קורא, רעיון ארכיטקטוני שגוי מתפשט בין ה-agents וחוזר שוב ושוב [1:02:24].

מה לא נאמר

- **הכיתוב ב-X לא מגובה:** בתמליל אין "delete your IDE", אין "of PRs at Google 85%", ו-Yegge לא מוצג כעובד Google נוכחי. הוא דווקא אומר ש-IDEs חוזרים בצורה חדשה [24:35].
- **אין מדדים על Gas Town:** לא נמסרו עלויות tokens, שיעורי הצלחה או זמני ריצה. המספר הקונקרטי היחיד הוא יותר מ-100 PRs בימים הראשונים אחרי השחרור [1:14:44].
- **רמות 1-3 ו-8 חסרות:** הפתיח מזכיר שמונה רמות ואומר ש-70% מהמהנדסים תקועים בתחתית. במקטעי התמליל שהתקבלו מתוארות רק רמות 4-7, ולנתון של 70% אין מקור.
- **"16 times smarter" לא מוסבר:** הוא מדבר על "two more cycles", בלי להגדיר מה זה מחזור ואיך מודדים "חכם יותר" [1:06:58].
- **הסיכונים של ה-swarm לא כומתו:** Yegge עצמו מזהיר ש-AI יכול להחמיר incident, אבל לא מציע שום מנגנון בקרה [38:27].

2. סולם האבסטרקציה: מקומפילרים ועד Claude Code

Yegge מתחיל מהקריירה שלו. מה שצריך לדעת כדי להיות מהנדס משתנה כל כמה שנים. שאלות הראיון על מניפולציית ביטים שהוא וחבריו כל כך אהבו "we were starting to bounce off candidates who'd never seen a bit" [4:38]. הוא מודה שחלק גדול מהאגו שלו היה קשור לרקע בקומפילרים, ושהידע הזה כבר לא שימושי "in any meaningful sense" [6:09]. ההסבר שלו פשוט: "We're just walking up the abstraction ladder" [6:09].

על AI הוא אומר שבהתחלה היה ספקן. ChatGPT ב-2022 כתב פונקציות "janky Emacs Lisp". כשהגיעו השמועות על כלי command line פנימי ב-Anthropic, התגובה שלו הייתה "just flat out rejection" [10:46]. מה ששכנע אותו היה השימוש עצמו. הוא מציין גם את הקביעה של Erik Meijer (ה-ASR כותב "Dr. Eric Meyer") שמפתחים יפסיקו לכתוב קוד, ומסביר אותה כך: "he sees the curves, man" [13:51]. לטענתו, מי שמאמין בעקומת S שכבר מגיעה לשיא טועה בתזמון.

Steve Yegge [10:46] And then I was like, oh, I get it. We're all doomed. Right? And then I wrote —
Death of the Junior Developer right after that, actually, I think

3. רמות האימוץ, ה-IDE ו-Claude Cowork

מהרמות ש-Yegge מתאר, נשמרו בתמליל רמות 4 עד 7. ברמה 4 האמון ב-agent גדל ומסתכלים פחות על ה-diffs. ברמה 5 "I'll look at the code in my IDE later, but I'm not coding with my IDE". ברמה 6 משעמם לחכות ל-agent אחד, אז מפעילים עוד אחד. משם קצרה הדרך להתמכרות. רמה 7 היא הבלגן: הודעה ל-agent הלא נכון, פרויקט שלם שנבנה בתוך פרויקט אחר. ומכאן השאלה שמובילה ל-Gas Town [23:03].

Steve Yegge [23:03] And so you fire up another agent and now you're addicted because you'll —
very quickly get into an equilibrium where every agent is waiting. There's always an agent waiting for
you because somebody's finished, right

על ה-IDE, בהמשך לדיון שלו עם Nathan Sobo מ-Zed (ה-ASR כותב "natan sobo"), Yegge אומר שה-IDE מעולם לא נועד באמת לכתיבת קוד. הוא נועד לחבר כלים לכלי אחד גדול, ואת התפקיד הזה ממלא היום ה-MCP. "[24:35] think Claude co-work is a return to the IDE form", וה-form factor הזה יתאים למפתח הממוצע יותר מ-Claude Code. Orosz מ-Craft Agents בשם Craft Agents עם ממשק ויזואלי, ושיש מפתחים שמעדיפים אותו.

הטענה הכי פרובוקטיבית בפרק הזה: "Claude Code 'makes you read a' [27:37]. Most people can't read" [29:10]. lot". חמש פסקאות שנראות למרבית האנשים כמו חיבור שלם הן מבחינת המודל "just clearing its throat" [29:10]. המסקנה שלו: עד שיגיעו UIs טובים, מי שלא קורא מהר יהיה בעמדת נחיתות, ונצטרך "recursive summarization".

מה זה אומר בפועל Yegge לא מציע למחוק את ה-IDE. הוא צופה שה-IDE יחזור כממשק שיחה ומעקב אחרי כמה agents. המסר בכיתוב של ה-repost הפוך למה שהוא אומר בפועל.

4. Gas Town: Mayor, Polecats ו-Crew

Gas Town אמור לעבוד כך: "one mayor that you talk to that's your your person", וה-Mayor מפעיל workers לפי הצורך [32:15]. בזמן ההקלטה הכלי היה "really broken all week" בגלל מעבר ל-Dolt (ה-ASR כותב גם "Dalt"), מסד נתונים עם יכולות של Yegge של git. מסביר: "beads is just git plus database crammed" [32:15]. את Dolt הוא הכיר דרך הקהילה אחרי ששחרר את Gas Town [1:14:44].

הרעיון המעניין ביותר הוא איך Gas Town מתמודד עם "minimaxing context argument" [32:15]. מחנה אחד ממלא את ה-context window עד הקצה כדי שהמודל יהיה "wise and all-knowing". המחנה השני מעדיף חלונות קצרים, בגלל העלות שגדלה בריבוע והירידה בתפקוד הקוגניטיבי כשה-context מתארך. Yegge משתמש בשתי הגישות לפי סוג העבודה.

Steve Yegge [33:46] well polecats are the min and crew are the max i have two fundamental role —
worker roles in gas town

- **Polecat (min)**: משימה מוגדרת היטב שמפורקת לתתי-משימות ועומדת בפני עצמה. היא עוברת ל-worker עם context קצר.
- **Crew (max)**: בעיית תכנון קשה שדורשת כמה שיחות. Yegge אומר לו "read all these docs and then we'll talk".
- **Monitoring roles**: חלק גדול מהמורכבות קיים "Yegge". "to tell opus four or five to be smarter". שזה ייעלם כשהמודלים יאומנו לעבוד "as a factory worker" [33:46].

Steve Yegge [33:46] i went out and built something that doesn't deliberately doesn't work it's too —
hard it's too hard for the models even opus 4.5 is barely enough

Yegge אומר במפורש: "i don't think people should be using it", חוץ ממטרות מחקר [38:27]. ובכל זאת הוא מביא מקרי שימוש שאחרים זיהו בחברות מסוג Fortune 50. הראשון הוא חברה שמקימה data centers מותאמים אישית. אצלה ההתקנה היא "three months of miserable button presses" עם acceptance criteria ברורים, ולכן אפשר להפעיל עליה swarm שיתכנס לפתרון. השני הוא חקירת production incidents. כאן Yegge עצמו מזכיר ש-[38:27] "AI can make it a lot worse". התוכנית הבאה שלו היא לחבר 100 מופעי Gas Town ולחקור federation, בהשראת [36:55] Moltbook. הוא לא קשור רגשית לפרויקט: "I think it'll be supplanted by something better within six months, if not sooner" [1:16:19].

— Steve Yegge [38:27] there's a class of problems emerging for which you can swarm them because you don't care that the results are messy

5. הצד האנושי והארגוני: שחיקה, value capture וחברות גדולות

התחזית הקשה ביותר בשיחה: בחברות גדולות יש מעין "חוגה" מ-0 עד 100 שקובעת כמה מהנדסים צריך לפטר כדי לממן tokens לשאר. לדבריו כולם מכוונים אותה בערך ל-50%, ו-"half your engineers don't want to prompt" [18:31] anyway. מנגד, AI מאפשר לקבוצות של 2, 5, 10, 20 ו-30 אנשים להתחרות בתפוקה של חברות גדולות. [20:01] "if you're anti-ai at this point it's like being anti this is the sun"

— Steve Yegge [41:58] there's a vampiric effect happening with ai where it gets you excited and you work really, really hard and you're capturing a ton of value

השחיקה מתבטאת בתנומות באמצע היום, בעייפות ובעצבנות. אנשי סטארט-אפים "try to load each other up" [41:58] with enough context to force the other one into a nap. מכאן Yegge עובר לשאלת הערך. אם מהנדס תפוקתי פי 100 ועובד שמונה שעות, "the company captured all of that value and that is not a fair capture exchange" [43:30]. הקיצון השני גם הוא לא טוב: מי שעובד 10 דקות ביום ולא מספר לאף אחד. Orosz מזכיר את תרבות ה-996 בסטארט-אפים של AI ואת ההצעה של Dario לפצות עובדים גם אחרי שעזבו [46:41].

בהשוואה בין Anthropic ל-Yegge, Google מתאר את Anthropic כ-hive mind שפועל "like a pure functional data structure", ברוח הספר של Chris Okasaki: מוסיפים ולא משנים, "yes and". אין spec ואין roadmap, יש prototype ש"everybody just gathers around... like a campfire" [48:17]. על Google הוא אומר שזו תובנה שמעולם לא אמר קודם: החדשנות מתה כש-Larry Page אמר "more wood behind fewer arrows".

— Steve Yegge [52:56] Before then, there was more work than people. And after that, there were more people than work. And so people started to fight over the work

מכאן נובעת ההרחבה: גם בחברות גדולות עם מהנדסים תפוקתיים מאוד, הארגון לא מסוגל לקלוט את העבודה. המהנדסים נתקעים בצווארי בקבוק ועוזבים. "we're looking at the big dead companies. We just don't know" [56:08]. Zendesk they're dead yet" מופיעה כדוגמה לחברה שלא הפכה לפלטפורמה ותתקשה מול לקוחות AI-native שרוצים API. על MCP הוא מוסיף. Anthropic, לדבריו, מצאה ש-"what works better than MCP is having the AI write its own API to call the MCP" [57:42]. הוא מודה שהוא לא עוקב מספיק אחרי התחום כדי לנבא.

6. קוד שאף אחד לא קורא: heresies, Bitter Lesson אישית

כשנשאל על "Yegge", agentive debt מציג מונח מפוסט שהוא עוד לא פרסם: heresy. בקוד שנכתב ב-vibe coding ואף אדם לא קורא, "an idea can take root among the agents that's incorrect". זו יכולה להיות ארכיטקטורה או זרימת נתונים שגויה שיוצרת impedance mismatch עם שאר הקוד. הן "have a tendency to grow and to come back and they're really hard to weed out", והיו כאלה גם ב-Gas Town [1:02:24].

VP: "when I'm working with a team of 80 agents, it's not very different from working with a team of 80 engineers" [1:05:28] הוא

מסתמך על The Bitter Lesson. הוא לא נוקב בשם הכותב, ומדובר כנראה ב-Rich Sutton. העיקרון: "don't try to be smart. Just try to be large". הוא מסכים שהעקומה בסוף היא עקומת S, אבל טוען שנשארו לפחות שני מחזורים.

— Steve Yegge [1:06:58] But I can tell you that there are at least two more cycles left in this. And that means they will be at least 16 times smarter than they are today

על תוכנה אישית: "all the software we have right now is garbage" [1:10:07]. בסדנת vibe coding ביסדיני מישו כתב לעצמו אפליקציית check-in לחברת תעופה, ו-Southwest חסמה אותו כבוט. כש-Jeffrey Emanuel עשה ל-fork Yegge, beads, עודד אותו: "Fork, fork, fork" [1:10:07]. מה שאי אפשר לשכפל לדעתו הוא קשרים אנושיים [1:11:38]. בצוותים מהירים צריך להיות "extremely loud and transparent" לגבי מה שעושים. בלי זה עבודה הופכת למושגת תוך שעותיים, כמו בסיפור על פיצ'ר שנבנה לפי בקשה מלפני שעותיים ובכל זאת עורר כעס [1:14:44]. כך הוא פעל גם עם Gas Town: שחרור מוקדם ו-"100 plus PRs the first couple of days" [1:14:44].

7. מה עושים עכשיו

ההמלצה המעשית של Yegge קצרה. למהנדס בחברה שעדיין עובדת עם Copilot הוא מציע חצי שעה ביום עם Claude Code. לחברה הוא מציע "make your token burn as high as your investors will let you go", כי "that token burn is your practice" [1:19:25]. לשאלה מה יקרה אם המודלים יפסיקו להשתפר הוא עונה שזה לא משנה.

— Steve Yegge [1:19:25] because the damage is done opus 4.5 made this officially an engineering problem we don't need you ai researchers anymore thank you

- **להתחיל ברמה 5–6 ולא להישאר ברמה 4:** לעבוד עם agent אחד דרך שיחה, ואז להוסיף agent שני. לשים לב לפיצול הקשב ולמה שקורה ב-repo.
- **להפריד בין שני סוגי עבודה:** משימות מפורקות ומוגדרות עוברות ל-worker עם context קצר. תכנון נעשה ב-session ארוך עם context עשיר.
- **לחפש בעיות שאפשר להפעיל עליהן swarm: acceptance criteria** ברורים, ותוצאה שמותר לה להיות "messy". לא להתחיל ב-production incidents.
- **לחפש heresies:** בקוד שאף אחד לא קורא, לבדוק מדי פעם את הארכיטקטורה ואת זרימת הנתונים, כי שם הטעויות חוזרות.
- **לנהל את האנרגיה:** האפקט הערפדי אמיתי גם אצל מי שמתלהב. כדאי להחליט מראש מה הגבול.
- **Debugging:** Yegge שם לב ש-agents משתמשים ב-printf ולא ב-[1:20:55] debuggers. כדאי לבדוק אם הם יעבדו טוב יותר עם tooling מסודר.

שאלה פתוחה	מה נאמר	מה חסר
האם 50% מהמהנדסים בחברות גדולות באמת ילכו?	אנלוגיית ה"חוגה" ותקציב tokens שמתחרה במשכורות [18:31]	נתוני פיטורים או עלויות tokens אמיתיים
האם orchestration עובד היום ב-production?	Gas Town הוא proof of concept שלא כדאי להשתמש בו [38:27]	שיעורי הצלחה, עלות לכל משימה, השוואה ל-agent יחיד
MCP או CLI/API?	Anthropic מצאה שעדיף שה-AI יכתוב API משלו [57:42]	Yegge מודה שהוא לא עוקב מספיק אחרי התחום
איך מזהים ומסירים heresy?	הן גדלות וחוזרות וקשה לעקור אותן [1:02:24]	שיטה, כלים או מדדים לזיהוי
מי לוקח את הערך מפרודוקטיביות פי 100?	החברה לוקחת הכול, וזה לא הוגן [43:30]	מודל תגמול קונקרטי (נאמר שיש לו תשובה, אבל היא לא נשמרה בתמליל)
האם עוד שני מחזורים הם אמת או השערה?	"at least 16 times smarter" [1:06:58]	הגדרה של מחזור ושל איך מודדים "חכם יותר"

8. מילון מונחים

מונח	פירוש
Agent	מודל שפה שפועל באופן עצמאי, מריץ כלים ומבצע משימת תוכנה
Orchestrator	שכבה שמתאמת כמה agents, מחלקת משימות ועוקבת אחריהן
Gas Town	ה-orchestrator בקוד פתוח של Yegge, בנוי כמו "עיירה" עם Mayor ו-workers
Mayor / Polecat / Crew	ב-Gas Town: הסוכן שמדברים איתו, worker למשימה קצרה עם context מינימלי, ו-worker לתכנון עם context מקסימלי
Beads	כלי מעקב משימות של Yegge, שהוא מתאר כ-git ומסד נתונים "דחוסים יחד"
Dolt	מסד נתונים עם ניהול גרסאות בסגנון git, ש-Gas Town עובר אליו
Context window	כמות הטקסט שהמודל רואה בבת אחת. ככל שהיא גדלה, העלות עולה והאיכות יורדת
Heresy	מונח של Yegge: רעיון ארכיטקטוני שגוי שמתפשט בין agents בקוד שאף אחד לא קורא
Swarm	הפעלה של agents רבים במקביל על בעיה, כשמוכנים לקבל תוצאה מבולגנת
MCP	Model Context Protocol, פרוטוקול לחשיפת כלים למודלים
Claude Cowork	מוצר של Anthropic עם ממשק גרפי. לדעת Yegge זו "חזרה לצורת ה-IDE"
Vibe coding / Agentic engineering	פיתוח שבו ה-AI כותב את הקוד והאדם מנחה ובודק
The Bitter Lesson	מאמר שלפיו גודל ומחשוב מנצחים ידע אנושי ייעודי. הכותב לא נזכר בשמו בשיחה
Value capture	השאלה מי נהנה מהערך שנוצר, העובד או החברה
996	עבודה מ-9 בבוקר עד 9 בערב, שישה ימים בשבוע
Ralph loop / Ralph Wiggum	טכניקה של Geoffrey Huntley (ה-ASR כותב "Jeff Huntley"): לולאה שמריצה agent שוב ושוב עד שעומד ב-acceptance criteria