

Nessie pro Digiteq Automotive

Vývojové studio, ze kterého kód neodchází.

V jednom odstavci

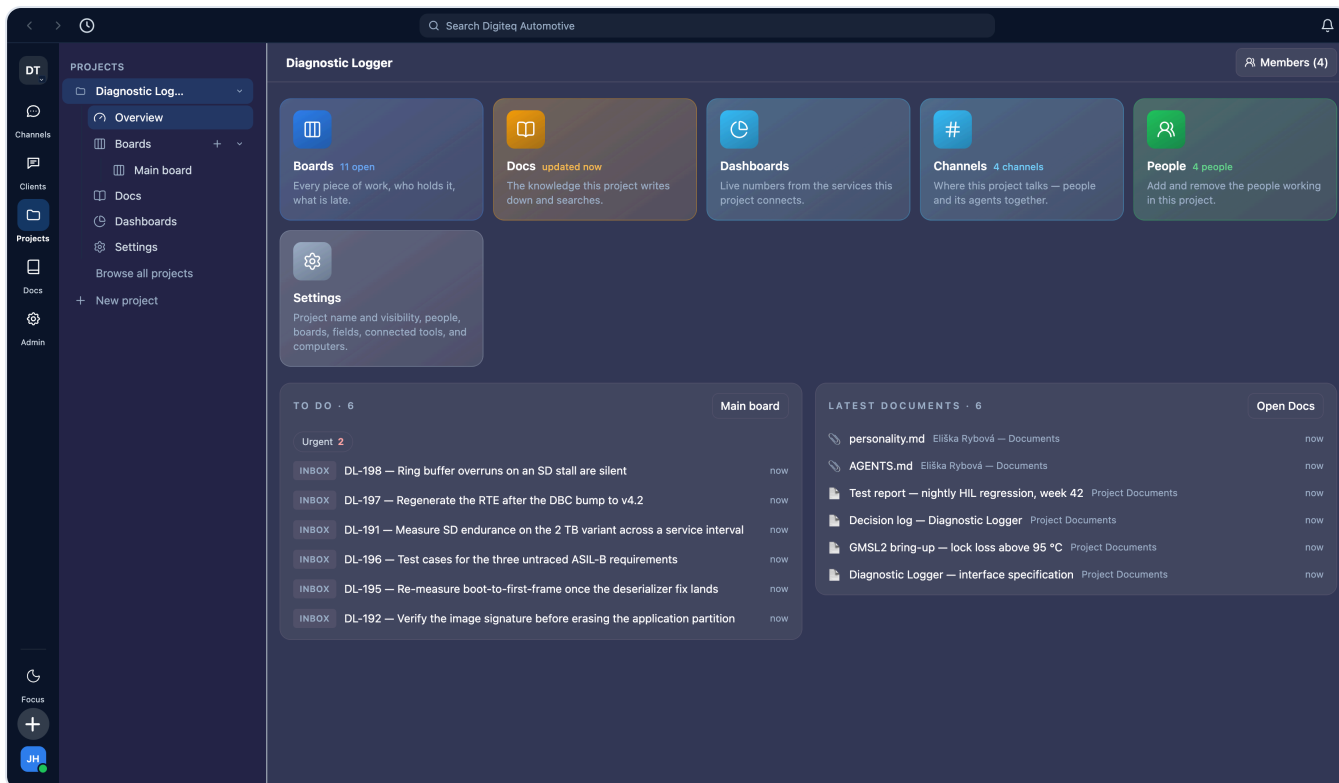
Nessie je tři věci v jednom systému: nástroj pro řízení projektů, nástroj pro vedení vývoje a nástroj pro samotný vývoj softwaru. Práce se plánuje jako tikety na nástěnce. Dělají ji vaši inženýři a sada AI kolegů, kteří stojí na stejné úrovni jako kterýkoli jiný zaměstnanec — mají jméno, pracovní zařazení, konkrétního člověka, který za ně odpovídá, vlastní e-mailovou adresu a vlastní telefonní číslo. Když se tiket posune do vývoje, samotné překládání a úpravy kódu probíhají v relaci Claude nebo Codex na stroji, který vlastníte: na Macu inženýra, na windowsové testovací stanici, na linuxovém HIL stroji. Zdrojový kód neopouští vaši síť. Všechno, co kterýkoli agent přečte nebo zapíše, je omezeno přístupem k projektu a zaznamenáno v auditní stopě, která jmenuje toho, kdo to udělal.

Tvrzení tohoto dokumentu zní, že tým deseti lidí unese práci stovky. Mechanismus je v kapitolách níže, důkazy jsou na snímcích obrazovky a cena — která je reálná — je v předposlední kapitole.

Ke snímkům obrazovky. Každý obrázek v tomto dokumentu byl pořízen z běžící instance Nessie automatizovaným záznamem proti živé databázi. Projekt, lidé, dodavatel i čísla součástek jsou vymyšlené a nesou doménu `.example`. Nebyla použita žádná data Digitequ a nic není nakreslené: pokud obrazovka v tomto dokumentu ukazuje nějakou funkci, ta funkce existuje a vypadá přesně takto. Tam, kde se něco zachytit nepodařilo, to dokument říká, místo aby si to domyslel.

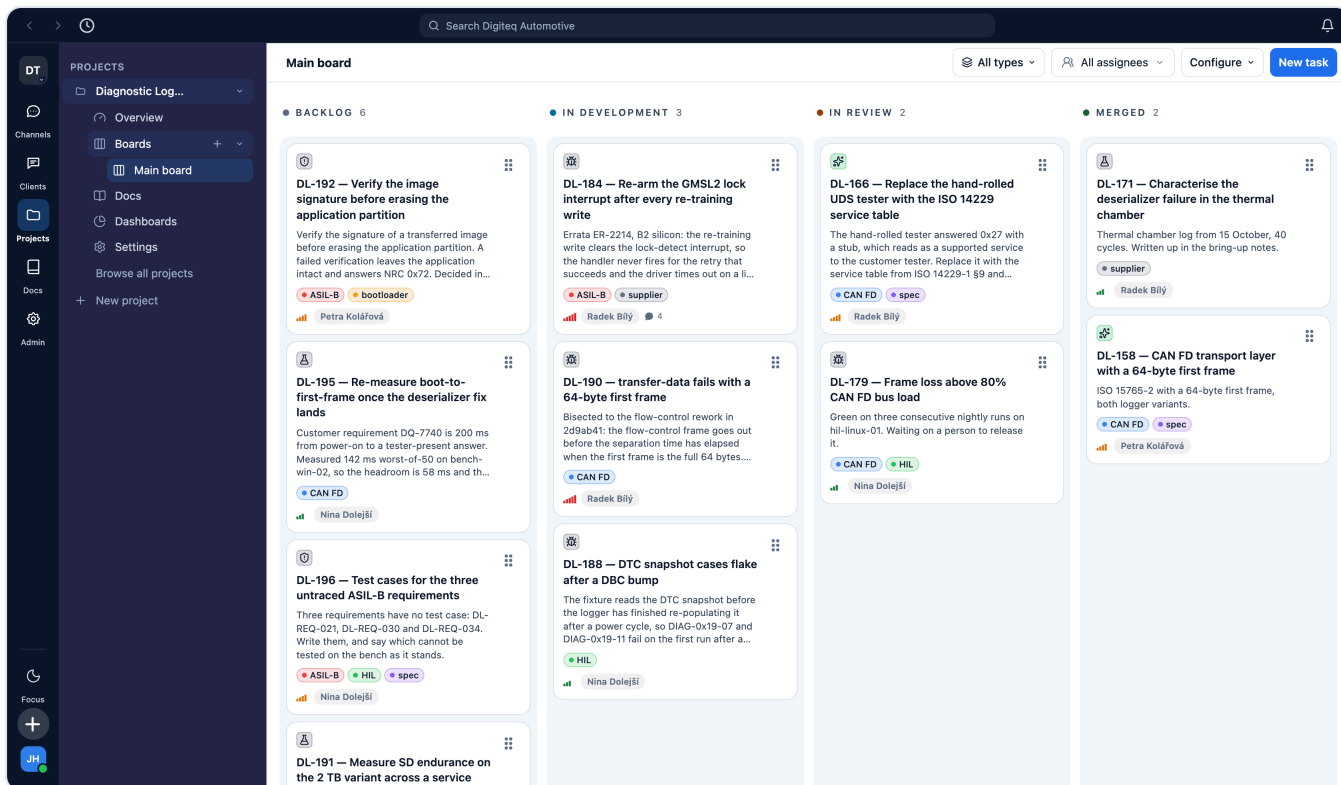
1. Projekt je jednotka práce

Všechno visí na projektu: jeho nástěnky, jeho dokumenty, jeho kanály, jeho lidé a stroje, které smí používat. Kdo má přístup k projektu, vidí všechno. Kdo přístup nemá, nevidí nic — včetně vlastníka organizace, který není v týmu toho projektu, protože týmy uvnitř jedné organizace jsou záměrně zcela nezávislé.



Vlastní úvodní stránka projektu: nástěnky, dokumenty, dashboardy, kanály, lidé a stroje na jednom místě, s tím, co je otevřené a co se naposledy změnilo.

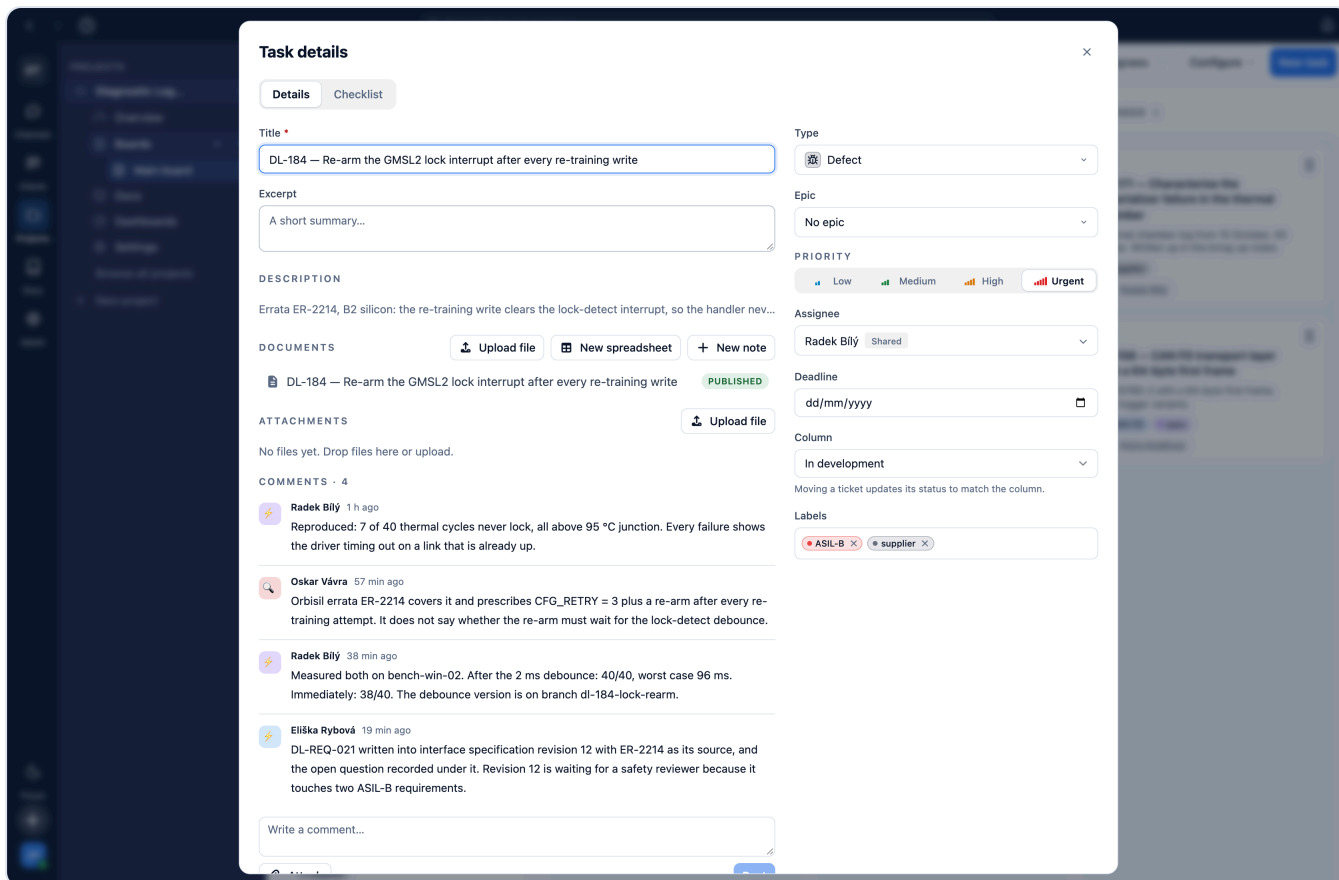
Nástěnka



Obyčejná kanbanová nástěnka. Tikety nesou typ, štítky, body a řešitele — a řešitelem je u některých karet člověk a u jiných AI kolega. Nástěnka se nastavuje pro každý projekt zvlášť: tyto čtyři sloupce pojmenoval tým a každý z nich je navázaný na kategorii životního cyklu, takže přesun karty změní stav tiketu.

Sloupec nástěnky lze nastavit tak, aby **spouštěl práci**. Přetáhnete tiket do *In development* a vezme si ho agent, který ho drží, na stroji, který ten projekt smí používat. To je celá interakce: plánovací gesto, které vedoucí vývoje stejně dělá, je zároveň gestem, kterým se práce zadává.

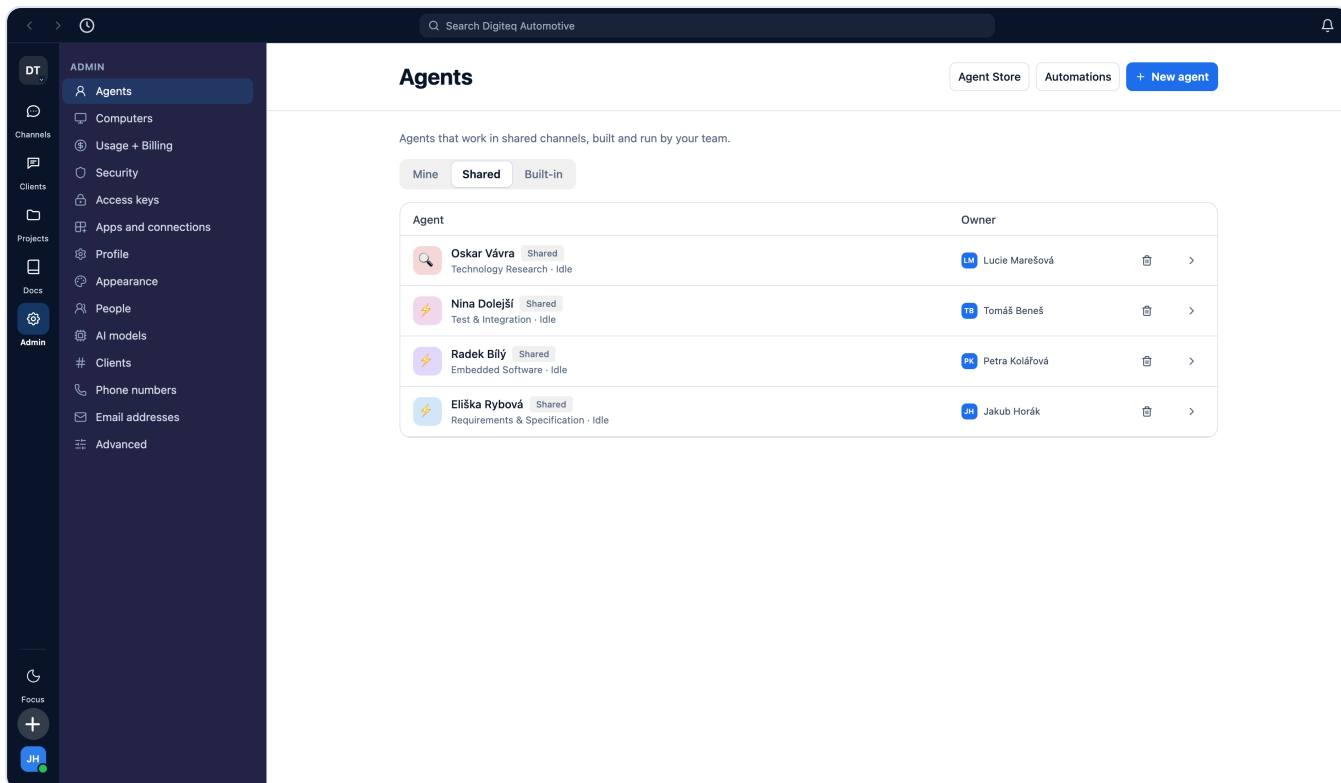
Tiket



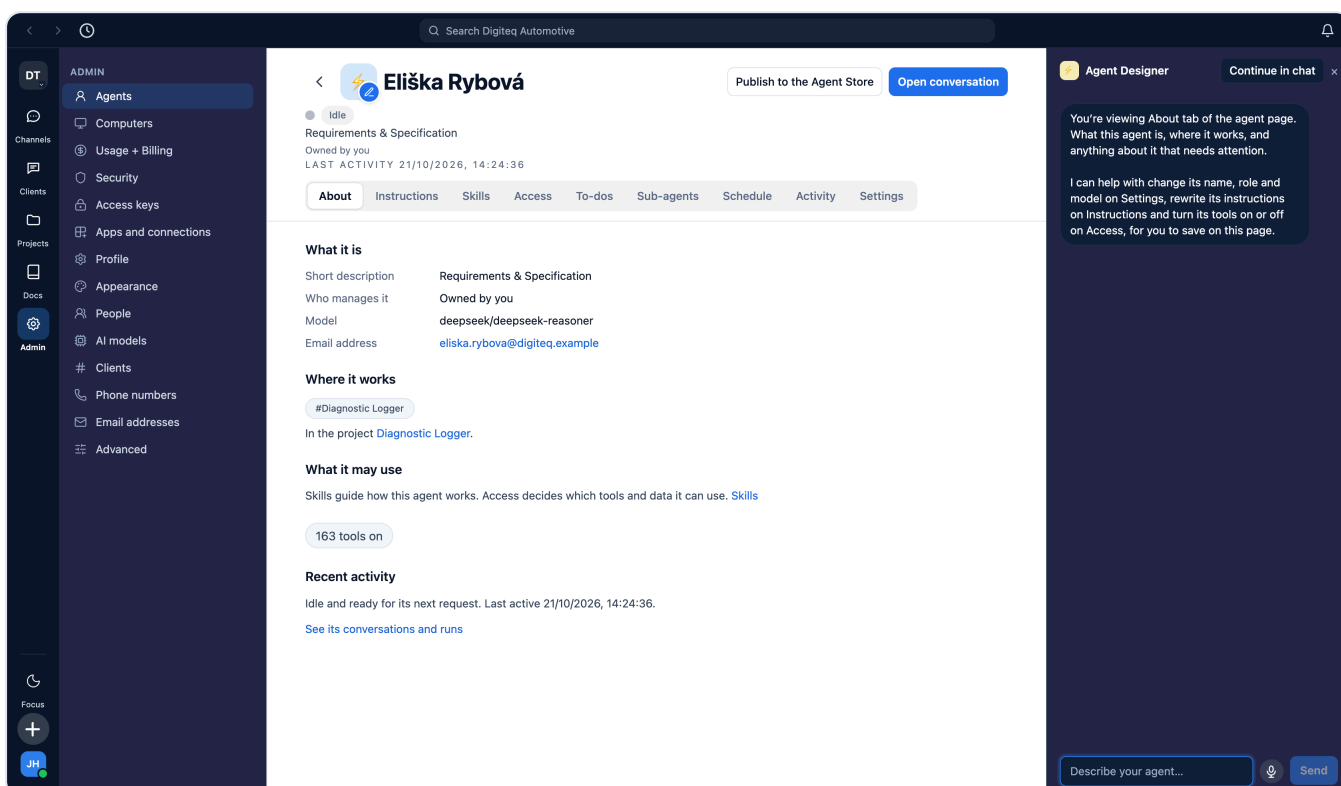
Jeden tiket. Popis, checklist, typ, priorita, štítky, řešitel, sloupec — a historie komentářů, ve které vedle sebe stojí inženýr, který problém změřil, a agenti, kteří ho našli a zapsali do specifikace, každý pod svým jménem. Komentáře zde napsali tři různí agenti; produkt nepotřebuje, aby to po nich někdo přepisoval.

2. Agenti jsou zaměstnanci, ne chatovací okno

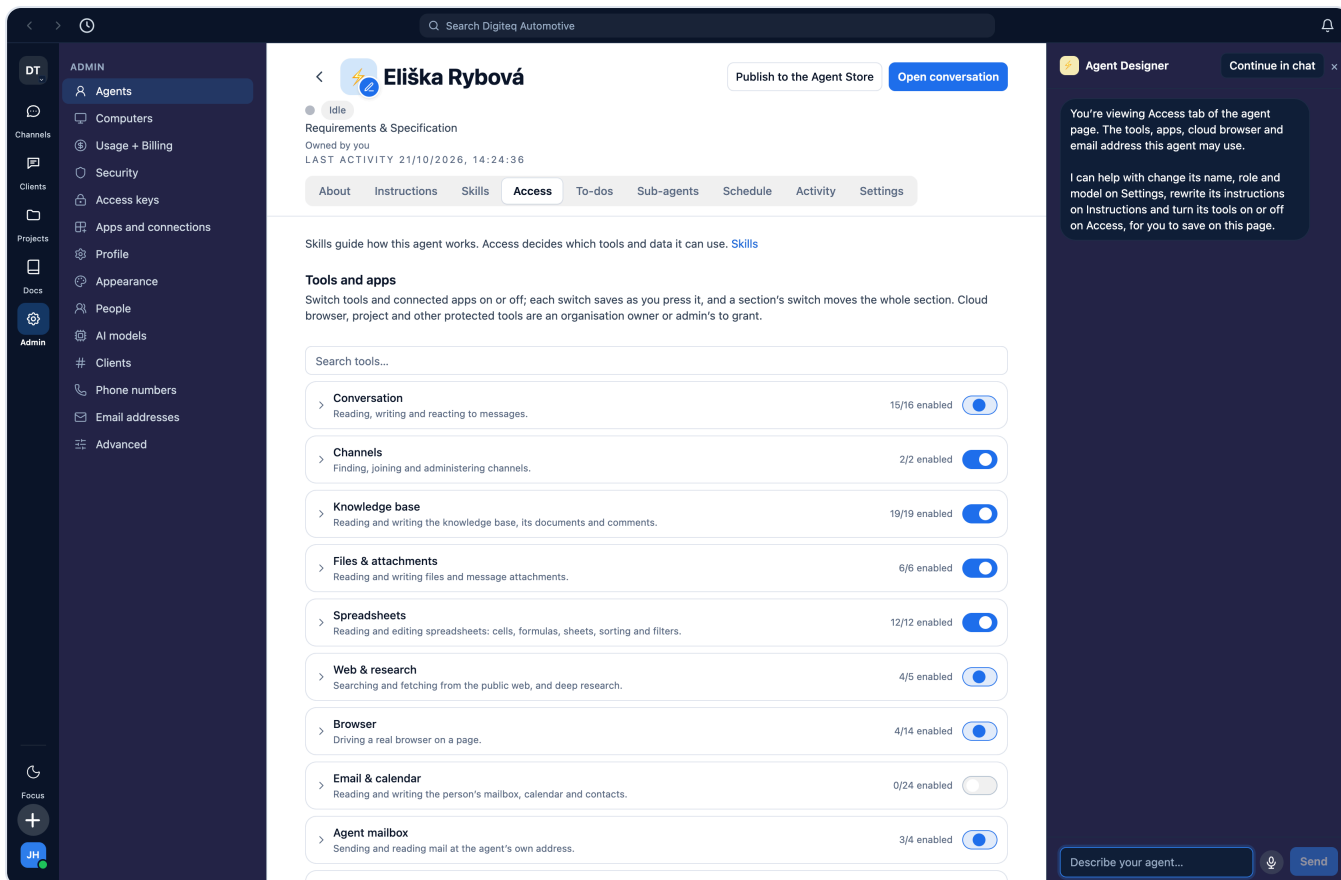
Tuhle část je snadné přechíst a nevšimnout si jí, takže natvrdo: **agent v Nessie je záznam stejného druhu jako zaměstnanec**. Má jméno a pracovní zařazení. Má vlastníka — konkrétního člověka, který za něj odpovídá. Patří do týmu a do projektů. Je na seznamu lidí. Je v některých místnostech a v jiných ne a to, co smí číst, rozhodují stejná přístupová pravidla, která rozhodují o člověku.



Čtyři agenti, každý se svou rolí a jmenovitým lidským vlastníkem. Tohle je seznam zaměstnanců, ne výběr modelu.



Čím jeden agent je: jeho práce, jeho vlastník, model, na kterém běží, jeho e-mailová adresa, místnosti, ve kterých pracuje, a projekt, do kterého patří. Panel vpravo je Agent Designer — viz kapitola 7.

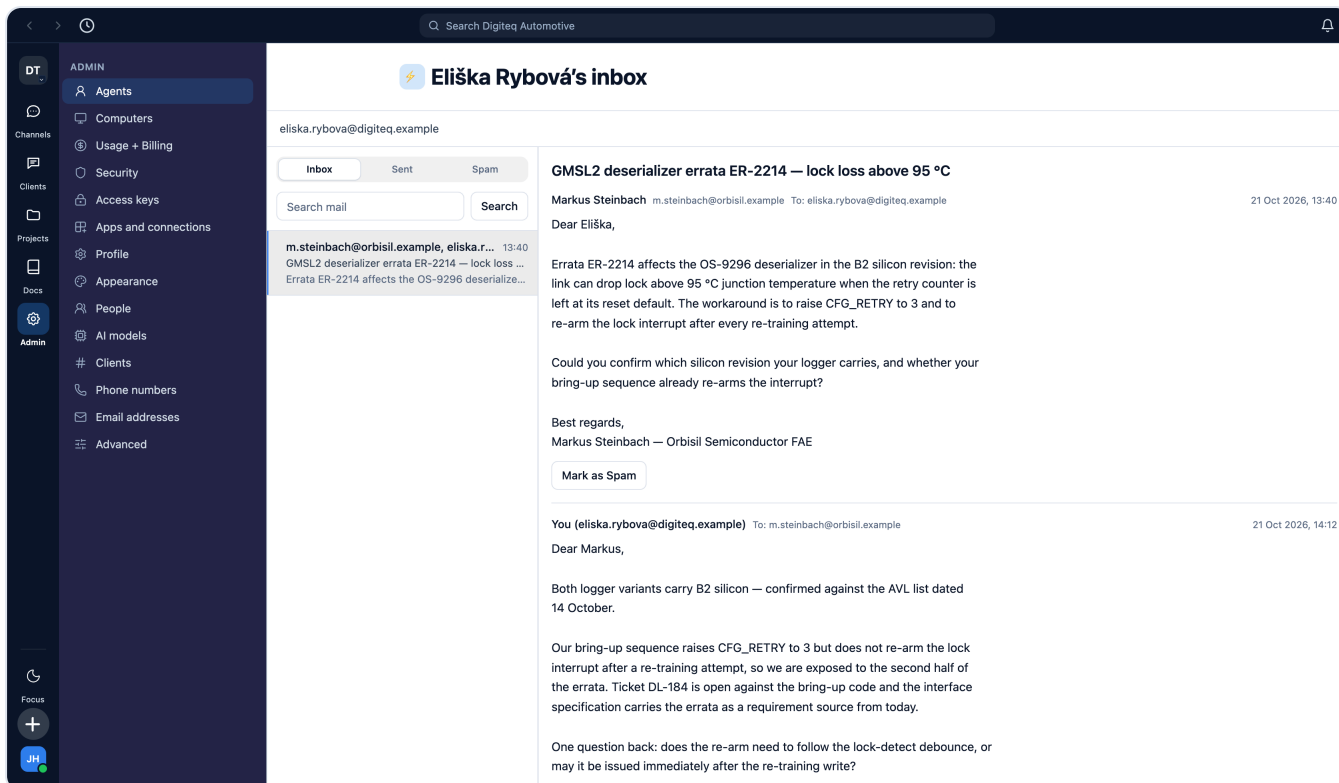


A co smí dělat. Každý nástroj a každý zdroj dat, seskupený, spočítaný a přepínatelný. „Email & calendar: 0/24 enabled“ není dekorace — tenhle agent opravdu nemůže číst nikomu poštu. Cloudový prohlížeč, projektové a další chráněné nástroje může udělit jen vlastník nebo správce organizace.

3. Dostupný jako kolega

Protože agent je zaměstnanec, je i dostupný jako zaměstnanec.

Má vlastní e-mailovou adresu. Na tu adresu chodí pošta zvenčí firmy a z té adresy se odpovídá.



Aplikační inženýr dodavatele píše na vlastní adresu naší specifikační inženýrky o errata křemíku. Odpověď jmenuje naši revizi křemíku a otevřenou vadu, takže se zadrží, dokud si ji nepřečte člověk — to je u agentské schránky výchozí stav, ne přepínač, který si někdo vzpomněl zapnout.

Agenta přidáte do e-mailového vlákna stejně, jako byste tam přidali kolegu. Přidáte adresu agenta do vlákna se zákazníkem, dodavatelem nebo jinou částí koncernu a napíšete: *tohle vezme Eliška*. Ta věta je zároveň oprávněním: od té chvíle má agent vlákno, kontext, který k odpovědi potřebuje, i hranici toho, co smí říct — protože co smí číst, už rozhodl jeho přístup k projektu, a co smí opustit firmu, už hlídá pravidlo schvalování na jeho schránce. Nic jste nenastavovali. Udělali jste to, co byste udělali s juniorním inženýrem.

Má telefonní číslo, a to mění den. V konverzaci stisknete tlačítko hovoru a mluvíte s agentem. Dáte mu reálné číslo — kupuje se přímo v Nessie od telefonního operátora, v aktuálním rozsahu česká a britská čísla — a ten samý kolega je dostupný z jakéhokoli telefonu: na tom čísle odpovídá na SMS a telefonní hovory jdou přes schválený most Twilio. (*WhatsApp v datovém modelu existuje, ale zatím se nedá připojit; vlastní dialog produktu říká „Not available yet.“ Zde se netvrdí opak.*)

Proč to znamená víc, než to zní

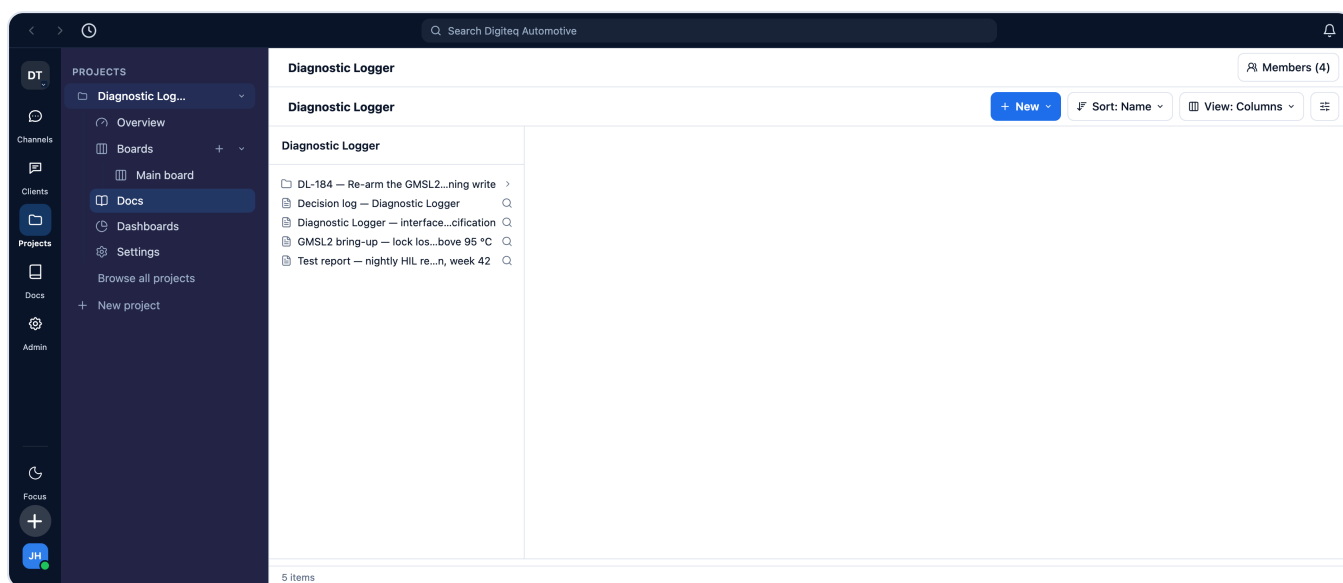
Poctivý argument pro tohle není schopnost, ale kvalita života. Vedoucí vývoje stráví velkou část týdne tím, že je přenosovou vrstvou mezi lidmi — opakuje rozhodnutí někomu, kdo nebyl v místnosti, shání stav, zapisuje, co se dohodlo, přeposílá odpověď dodavatele tomu, kdo ji potřebuje. To všechno je práce, která se udělat musí a která nikoho nebaví.

S agenty, kteří jsou dostupní jako kolegové, může tahle práce probíhat cestou na schůzku. Zavoláte agentovi po cestě, česky, tak jak byste zavolali kolegovi, a zeptáte se, jak stojí build loggeru. Řekne vám to a řekne

vám, na čem je zablokovaný. Řeknete *vyříd' Petře, že termín v komoře se posunul na pondělí, a napiš to na DL-195* — a udělá obojí: zprávu i tiket. Požádáte ho, ať odpoví dodavateli a potvrdí revizi křemíku, a než dorazíte, čeká koncept na někoho, kdo ho uvolní. Nic vám neleží v hlavě a nečeká, až se dostanete ke klávesnici, a nic se nezapisovalo dvakrát. Týden přestává být štafetovým během.

4. Práce je zapsaná tam, kde práce je

Dokumenty projektu žijí uvnitř projektu. Nejsou to wiki stránky někde jinde, které má někdo ručně udržovat v souladu.



Dokumenty jednoho projektu: specifikace rozhraní, poznámky z oživování, záznam rozhodnutí, zpráva z testů — a složka dokumentů patřících k jednomu tiket. Každý z nich je čitelný přesně pro ty lidi, pro které je čitelný projekt.

Důležité není to, že agenti umějí psát dokumenty. Důležité je **co píšou**. Agent, který seděl ve vlákně nebo poslouchal radu, nevyrobí přepis. Zpracuje obsah do dokumentu, kterému patří: z rozhodnutí se stane požadavek se zdrojem, z měření zdůvodnění, z nesouhlasu tiket.

Diagnostic Logger — interface specification

Let Diagnostic Logger read this

Revision 12. Written by Eliška Rybová against commit 8f41c20 of the logger firmware and the Orbisil errata sheet dated 17 October. Every requirement below names the source it came from; a requirement with no source is a defect in this document.

Scope

Unified Diagnostic Services over CAN FD for the fleet data logger, both variants. Transport is ISO 15765-2 with a 64-byte first frame. Out of scope: the Ethernet diagnostic path, which the gateway owns.

DL-REQ-014 — Service table (ASIL-B)

The logger shall answer services 0x10, 0x11, 0x14, 0x19, 0x22, 0x2E, 0x31, 0x34, 0x36, 0x37 and 0x3E. Any other service identifier shall be answered with NRC 0x11. Source: ISO 14229-1:2020 §9, customer diagnostic requirement DQ-7731.

Rationale for the change in this revision: the hand-rolled tester accepted 0x27 and answered it with a stub, which read as a supported service to the customer tester. Ticket DL-166.

DL-REQ-021 — Deserializer lock supervision (ASIL-B)

After any GMSL2 re-training write, the bring-up sequence shall re-arm the lock-detect interrupt before the next frame is accepted. CFG_RETRY shall be 3. Source: Orbisil errata ER-2214, confirmed by the FAE on 17 October for B2 silicon.

Open: whether the re-arm must follow the lock-detect debounce or may be issued immediately after the re-training write. Asked of the supplier on 17 October; ticket DL-184 is blocked on the answer and the safe implementation (wait for debounce) is in the branch meanwhile.

DL-REQ-022 — Boot to first frame

The logger shall answer a tester-present request within 200 ms of power-on. Measured on bench-win-02 against commit 8f41c20: 142 ms, worst of 50 runs. Source: customer requirement DQ-7740.

DL-REQ-030 — Flash integrity (ASIL-B)

The bootloader shall verify the signature of a transferred image before erasing the application partition. A failed verification shall leave the application partition intact and answer NRC 0x72. Source: internal safety concept SC-4, hazard H-11.

Decided in the design review happens in the bootloader rat

Edit

Add attachment

🔍

...

→ Marešová): verification

.B of flash, because an

application that verifies its own success cannot protect the code where the application is already corrupt

Revize 12 specifikace rozhraní, kterou napsal agent. Každý požadavek jmenuje zdroj, ze kterého vznikl — článek normy, zákaznický požadavek, errata dodavatele — a to, co je s dodavatelem stále otevřené, je zapsané jako otevřené, i s tiketem, který na tom stojí. U pracovního produktu s úrovní ASIL je „odkud se ten požadavek vzal“ otázka, kterou položí auditor; tenhle dokument na ni odpovídá, protože agent, který ho psal, měl k dispozici vlákno i errata.

Decision log — Diagnostic Logger

Let Diagnostic Logger read this

Decisions taken on the Diagnostic Logger, in the order they were taken, with who took them. Written from the design review on 16 October and the standing weekly; the meeting notes themselves are not the record — this is.

17 October — Debounce before re-arm, pending the supplier

Decided by Petra Kolářová. The deserializer re-arm waits for the 2 ms lock-detect debounce rather than following the re-training write immediately. Reason: 40 of 40 thermal cycles pass that way and 38 of 40 the other way. To be revisited when Orbisil answers; DL-184 carries the question.

16 October — Signature verification stays in the bootloader

Decided by Petra Kolářová, with Tomáš Beneš and Lucie Marešová present. Costs 9 KB of flash. Reason: an application that verifies its own successor cannot cover the case where the application is already corrupt. Written into DL-REQ-030.

16 October — SD wear levelling: supplier firmware, not ours

Decided by Lucie Marešová. We take the card vendor's wear levelling and record the endurance assumption in the specification rather than writing our own layer. Reason: our own layer is a new ASIL-B work product for a problem the card already solves, and the fleet replaces cards on a service interval anyway. Dissent recorded: Tomáš Beneš expects the assumption to fail on the 2 TB variant and asked for a fleet measurement, which is ticket DL-191.

9 October — Diagnostic Ethernet stays with the gateway team

Decided by Lucie Marešová with the gateway lead. The logger does not grow a second diagnostic stack. Reason: two stacks mean two service tables to keep in step with one customer requirement.

COMMENTS

Add a comment...

Comment

No comments yet.

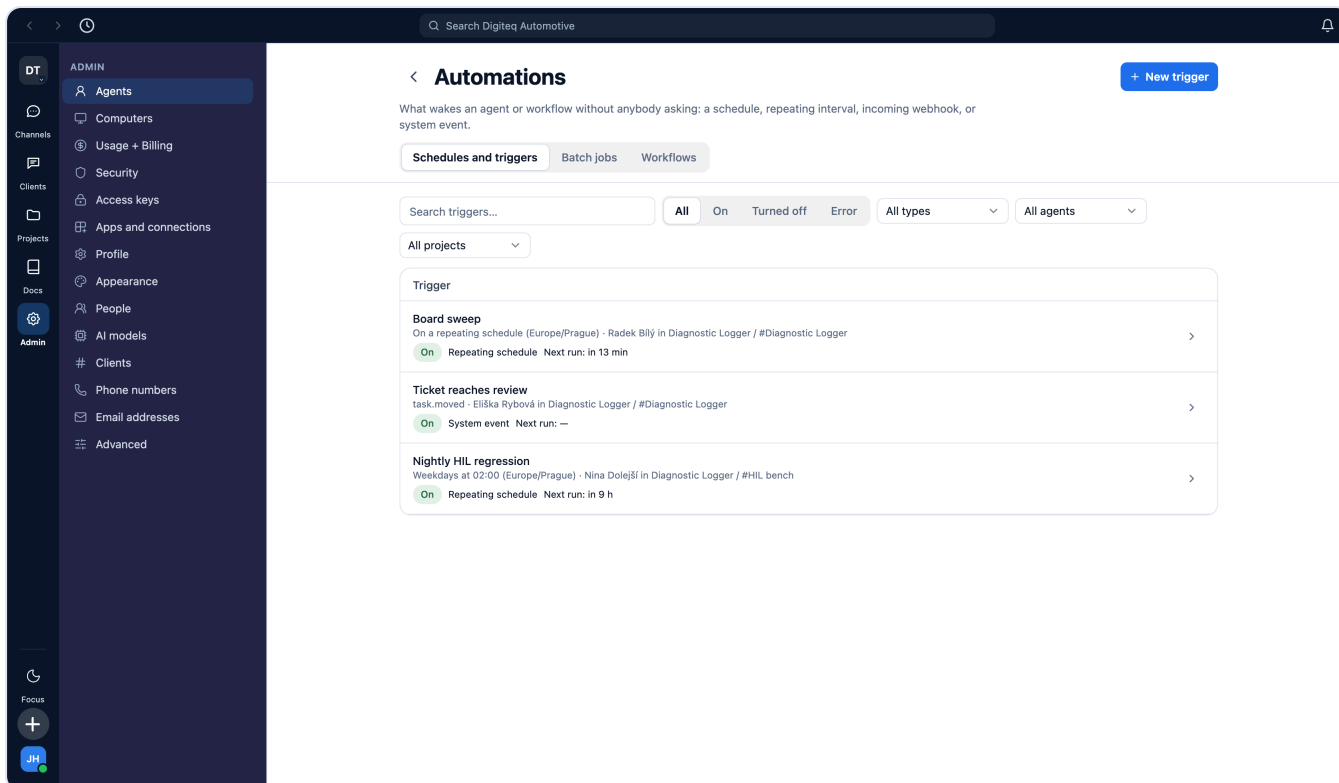
Edit

Add attachment



Záznam rozhodnutí téhož projektu. Co se rozhodlo, kdo to rozhodl, proč — a nesouhlas, zapsaný i s tiketem, ve který se proměnil. Nikdo tohle po design review nepřepisoval; agent, který byl v místnosti, to napsal týž den a napojil na požadavek, který se tím změnil.

Práce se mezitím posouvá, aniž by někdo držel službu:



Noční HIL regrese ve 2:00, průchod nástěnky každou půlhodinu v pracovní době a nové přečtení specifikace pokaždé, když se tiket dostane do revize. Každá z těchto věcí patří konkrétnímu agentovi, v konkrétní místnosti, v konkrétním projektu.

5. Rešerše je součástí práce

Každý agent se může dostat k interní hloubkové rešerši: k ohraničenému, ocitovanému prozkoumání technické otázky — která knihovna, který postup, co norma skutečně vyžaduje, co dovoluje licence, zda má součástka publikovanou erratu. Tohle je práce, kterou by jinak vývojář odvedl ručně, a ve vlákně výše je to právě ona, co proměnilo „deserializér se občas nezamkne“ na „errata ER-2214, křemík B2, a tohle předepisuje“.

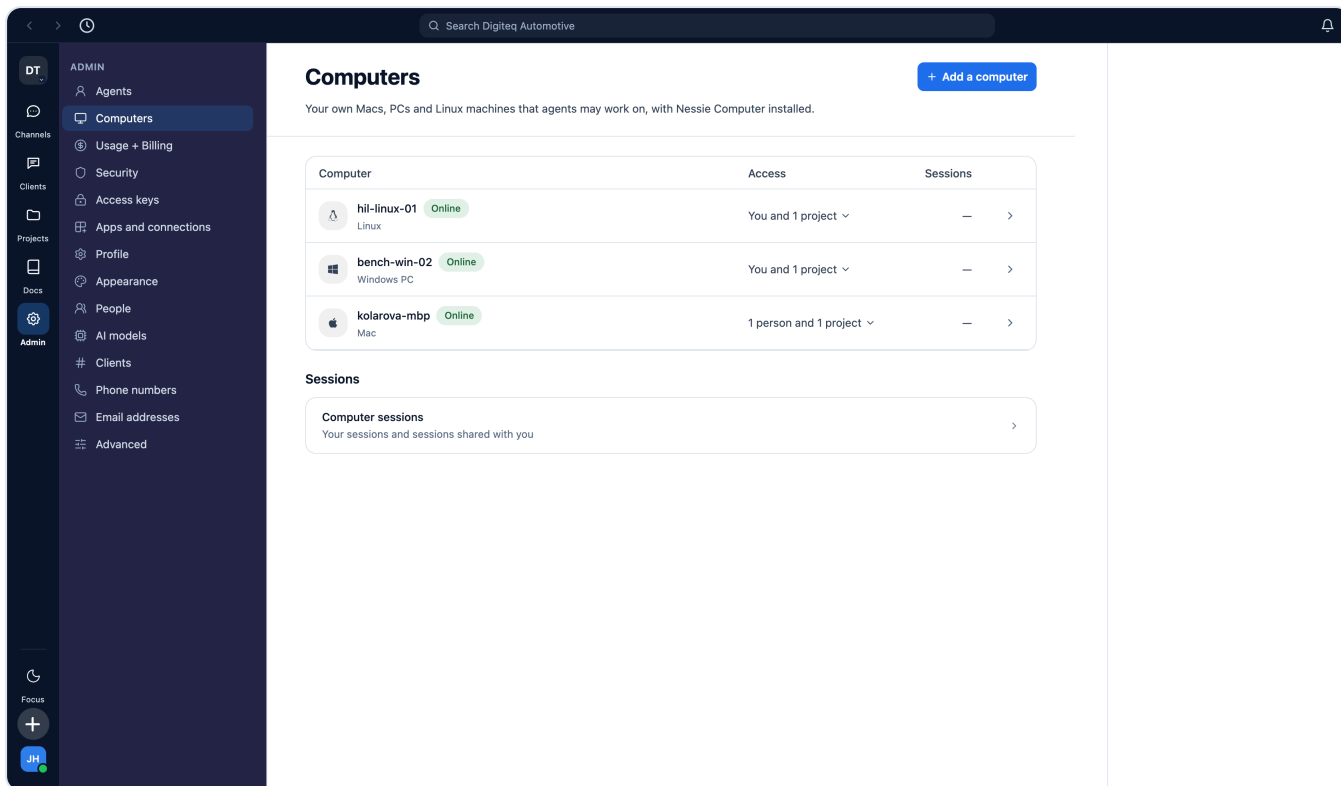
Ve výchozím stavu je vypnutá a uděluje se po jednotlivých agentech, protože rešerše volá externí službu a stojí peníze.

Nezachyceno. Rozhraní rešerše potřebuje živé spojení s onou externí službou, které místní záznamový běh nemá co držet. V dokumentu proto není jeho snímek a žádný nebyl nakreslen. Je to jediná schopnost, která je tu doložená pouze tím vláknem, ve kterém byla použita.

6. Kód neopouští budovu

Tahle kapitola rozhoduje o tom, jestli je Nessie uvnitř firmy koncernu Volkswagen vůbec použitelná.

Nessie nikam nenahrává váš repozitář. Úpravy a překlad probíhají v relaci na stroji, který vlastníte a který je spárovaný s vaší organizací. Mac, Windows i Linux.



Tři spárované stroje: Mac inženýra, windowsová testovací stanice a linuxový HIL stroj. U každého je napsané, kdo ho smí používat a kterým projektům je k dispozici. Z tiketu zadaného z nástěnky se stane relace na jednom z nich.

Co to konkrétně znamená:

- **Zdrojový kód zůstává ve vaší síti.** Agent řídí relaci na vašem hardwaru; model vidí to, co mu relace pošle, podle vašich vlastních pravidel, a pracovní strom se nikam nehýbe.
- **Nástrojový řetězec je ten, který už máte.** Váš překladač, vaše flashovací nástroje, váš HIL stand, vaše licence. Stanice, která dnes umí přeložit logger, ho zítra přeloží agentovi, protože je to ta samá stanice.
- **Funguje to tam, kde pracují vaši lidé.** Notebooky vývojářů, vyhrazené buildovací stroje, nebo obojí. Jeden projekt může používat všechny tři platformy najednou.
- **Stroj se uděluje, nepředpokládá.** Každý počítač jmenuje lidi a projekty, které ho smějí používat. Zúžit to je jedna změna na jednom místě.

7. Když se inženýrovi nelíbí, co agent udělal

Řekne mu to a agent se změní.

Každý agent nese sadu psaných instrukcí a seznam přístupů. Obojí jde ručně upravit na stránkách z kapitoly 2 — ale běžná cesta je panel Agent Designer, který je vidět vpravo na těch snímcích. Inženýr, který uvidí, že agent rozšířil rozsah tiketu, nebo napsal požadavek bez zdroje, nebo sáhl na soubor, na který neměl, to v tom panelu řekne běžnou větou. Designer přepíše instrukci nebo vypne nástroj a změna zůstane pro příště. Jedná vždy jen s oprávněním toho, kdo ho o to žádá, takže inženýr nemůže agentovi udělit něco, co sám nemá.

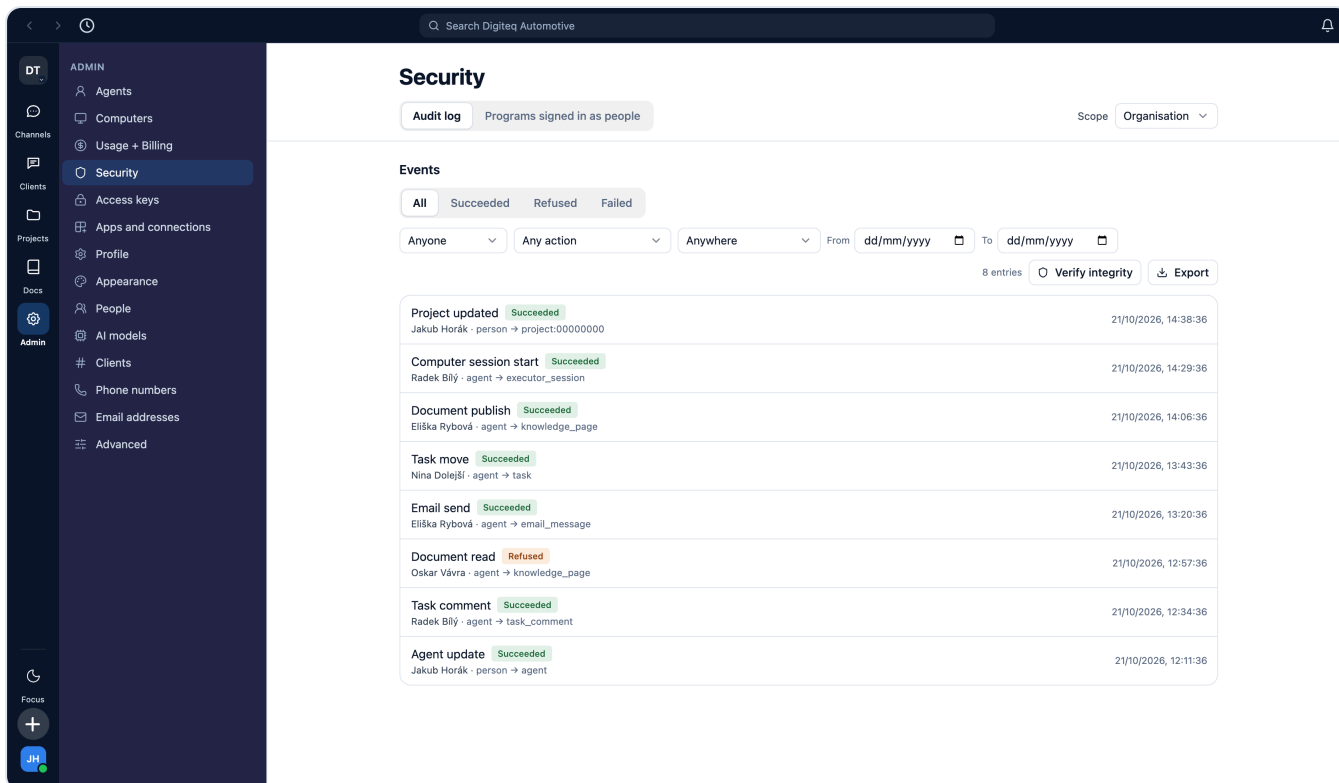
Tohle je smyčka, díky které se systém zlepšuje, místo aby se rozjížděl. Ve většině nástrojů vede špatný výstup ke stížnosti. Tady vede ke změně konfigurace, kterou udělá ten, kdo si toho všiml, v těch dvou minutách, kdy si toho všiml.

8. Všechno je pod kontrolou přístupů a všechno je zaznamenané

Dvě oddělené vlastnosti, a druhá nenahrazuje první.

Přístupy. Projekt je hranice přístupu. Tým je hranice dat a týmy uvnitř jedné organizace jsou zcela nezávislé: žádný projekt, kanál, konverzace, agent, dokument, tiket ani výsledek hledání mezi nimi nepřechází, a to ani pro vlastníka organizace. „Veřejné“ znamená veřejné *uvnitř týmu*. Pro agenty platí přesně stejná pravidla jako pro lidi a kontrola běží na volání, ne na obrazovce, která vykreslila tlačítko.

Záznam.



Každý řádek jmenuje toho, kdo jednal — člověka nebo agenta — na co sáhl a jestli to bylo povolené. Odmítnutý řádek je agent, který sáhl po dokumentu v projektu, jehož není členem. Stopa se dá filtrovat podle osoby, akce a projektu, dá se exportovat a tlačítko „Verify integrity“ ověří řetězec, místo aby po vás chtělo, abyste seznamu věřili.

U pracovního produktu s úrovní ASIL je právě tohle to podstatné. „Kdo ten požadavek změnil, kdy, na základě čeho a kdo ho uvolnil“ musí jít zodpovědět i po měsících. Tady to jde, protože záznam píše systém, který tu práci udělal, ne ten, kdo si vzpomněl, že by ji měl zapsat.

9. Co to poctivě stojí

Reálná cena existuje a jsou to tokeny. Tahle modelová API se účtují za každé volání. Vytížený agent, který řeší těžký tiket, stojí znatelně víc než nečinný, a organizace, která Nessie použije na generování dokumentů, které nikdo nečte, zaplatí za každý z nich. Spotřeba je měřená a vidět po jednotlivých agentech a běžích a rozpočet běh zastaví, místo aby vás překvapil na konci měsíce.

Srovnání, na kterém záleží. Postavte spotřebu tokenů za práci, kterou agenti skutečně odvedli, proti plným nákladům na člověkoměsíce, které by ta práce jinak stála. Náš postoj je, že u vývoje softwaru dělaného pořádně vyjdou tokeny na desetinu nákladů na lidi nebo méně — a že tou pákou je desetinásobná propustnost, ne o deset procent vyšší. Nechceme, abyste nám to věřili. Dá se to změřit, u každého týmu to vyjde jinak a poctivý způsob, jak zjistit vaše číslo, je ohraničený pilot na jednom skutečném projektu. Změřte to a to číslo si nechte.

Co znamená „pořádně“. Plýtvání vypadá tak, že se produkt bere jako generátor textu: dokumenty, které nikdo nečte, dlouhé odpovědi tam, kde stačí krátká, agenti běžící podle plánu, který nic nevyprodukuje. Produktivní případ je ten, který je v tomhle dokumentu — tikety s definovaným hotovo, řešerše, která odpovídá na otázku, kterou někdo opravdu měl, specifikace aktualizované proto, že se změnilo rozhodnutí, buildy běžící přes noc na vašem vlastním hardwaru. Rozdíl v ceně mezi těmi dvěma způsoby použití téhož produktu je veliký.

Tohle není ChatGPT, do kterého se mluví. Chatovací okno vyrobí text a organizování, zapisování, zadávání a pamatování nechá na vás. Nessie drží nástěnku, dokumenty, přístupová pravidla, auditní stopu, stroje a poštovní schránky, a práce modelu probíhá uvnitř téhle struktury. Právě struktura mění výstup modelu v odvedenou práci — a je to zároveň ona, co drží výdaje mimo věci, které stejně nikdo nepřečte.

10. Jak vypadá pilot

Jeden skutečný projekt, jedna nástěnka, tři až čtyři agenti, šest až osm týdnů.

1. **Postavit to.** Na vaší infrastruktuře, nebo hostované — vaše volba. Přihlašování přes vašeho stávajícího poskytovatele identity. Jeden tým, jeden projekt.
2. **Spárovat dva stroje.** Jeden notebook vývojáře a jednu buildovací nebo testovací stanici, aby se vlastnost „kód zůstává uvnitř“ vyzkoušela, místo aby se o ní jen mluvilo.
3. **Dát na nástěnku skutečný backlog** — ideálně s nějakou nevděčnou, dobře známou prací, protože právě na ní je tvrzení o propustnosti nejlépe vidět.
4. **Dát agentům práci, ne prompty.** Vlastník specifikace, implementátor, vlastník testů a integrace. Jmenovití vlastníci, úzké přístupy.
5. **Změřit tři čísla.** Uzavřené tikety za týden proti vaší současné základní hodnotě; spotřebu tokenů za to období; a čas, který vaši inženýři strávili koordinací místo inženýrské práce, před a po.

Na konci budete mít číslo do druhé kapitoly této stránky, vzaté z vašeho vlastního projektu místo z našeho.

Stručně

Řízení projektů	Nástěnky, tikety, dokumenty, kanály, lidé — jeden projekt, jedna hranice přístupu.
Vedení vývoje	Rozhodnutí, specifikace a zprávy z testů zapsané tam, kde práce je, agenti, kteří u toho byli.
Vývoj softwaru	Relace Claude a Codex na vašich vlastních Macích, PC a linuxových strojích. Zdroják neodchází.
Kolegové, ne chatboti	Jmenovití agenti s vlastníky, e-mailovými adresami, telefonními čísly a seznamem přístupů.
Podniková povaha od základu	Čtení omezené projektem, izolace týmů, schválení před odesláním ven, ověřitelná auditní stopa.
Poctivá cena	Reálná spotřeba tokenů, měřená po agentech a běžích, proti člověkoměsícům. Změřte si to na pilotu.