



Nýjar leiðir í hugbúnaðargerð með gervigreind

Skref í átt að nýta gervigreind í hugbúnaðarþróun



Hefðbundin hugbúnaðarþróun í dag

Teymaskipting

Backenda teymi vinna með gagnagrunna og API, á meðan Frontenda teymi einbeita sér að notendaviðmóti. Hver hópur vinnur oft í sínu horni með takmarkaða sýn á heildarmyndina.

Agile vinnuferli

Sprintar eru venjulega 2–4 vikur þar sem teymi velja features úr backlog. Hver eiginleiki getur tekið daga eða vikur að þróa, prófa og skjala.

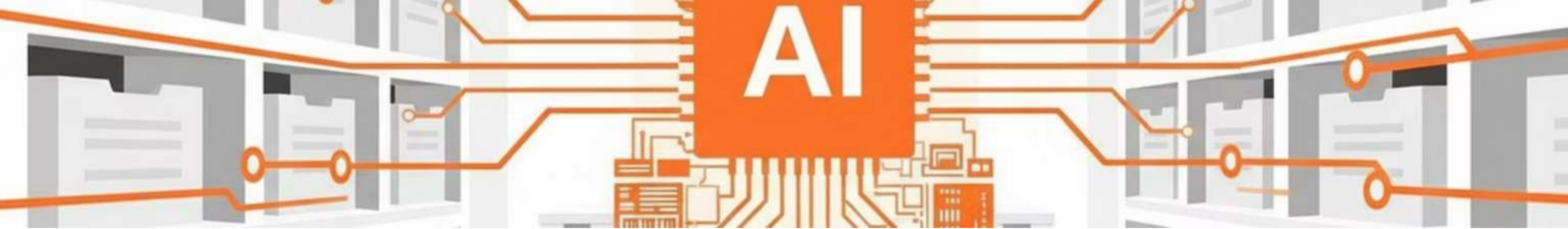
Samskipti og samhæfing

Upplýsingaflæði milli teyma byggist á fundum, tæknilegum skjölum og kóðaskoðun. Samhengi og samvirkni oft takmörkuð.



Lífsferill hugbúnaðarþróunar - SDLC





Context Engineering: Grunnurinn að árangursríkri AI-þróun



Eldri kóðasöfn

Búa til skipulag og skjölun inn í nákvæmar samhengisskrár svo AI-verkfæri eigi auðveldara að skilja eldri kóða og arkitektúr



Ný verkefni

Setja upp góðar venjur frá byrjun með skýrum reglum, samræmdum ferlum og uppbyggingu samhengis fyrir AI-verkfærin



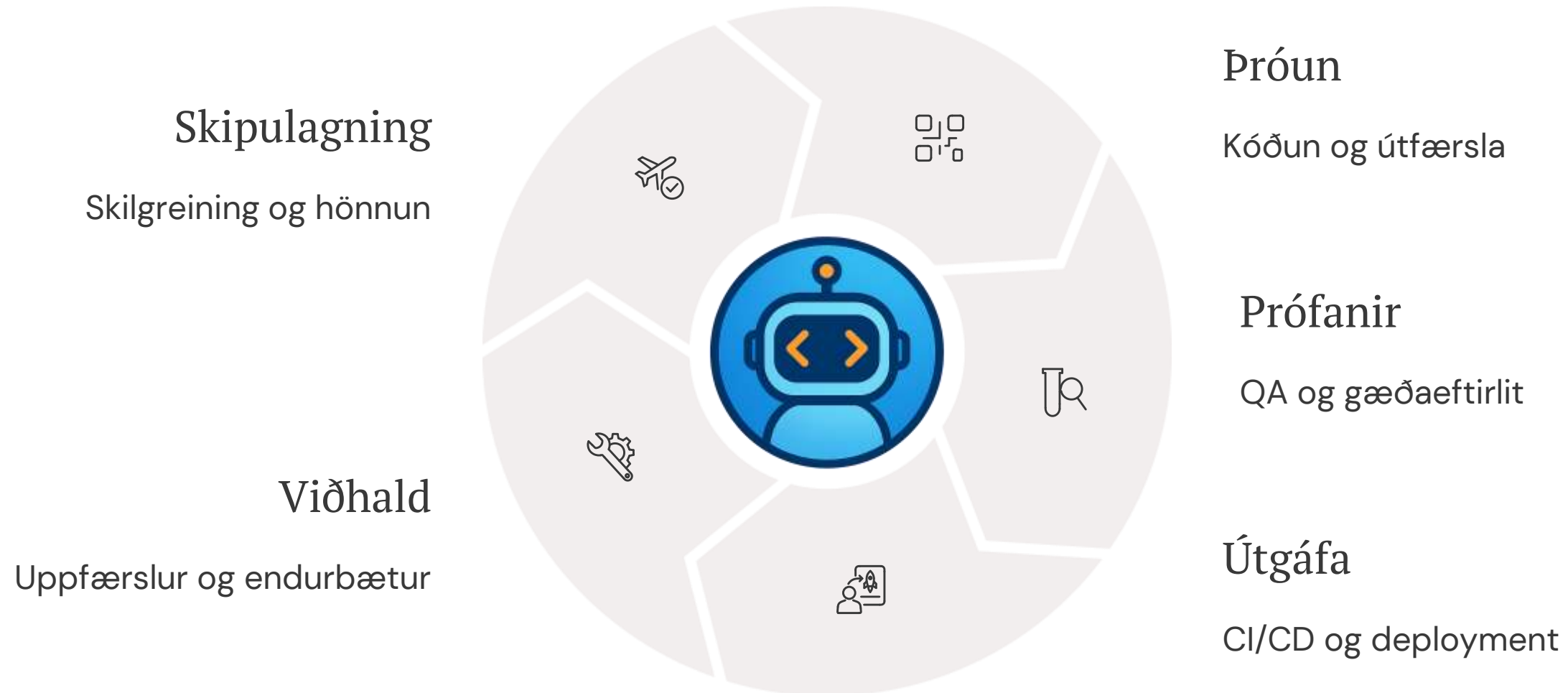
Stjórnun agenta

Búa til sérhæfða aðstoðarmenn sem þekkja þinn kóða og hægt að útvista auðveldum verkefnum á agenta.

 **Lykilatriði:** Samhengi (Context) er konungurinn – án góðs samhengis geta AI-verkfæri ekki veitt nákvæmar tillögur eða skilið kóðasafnið þitt.



Lífsferill hugbúnaðarþróunar - SDLC





AI í skilgreiningu lausnar og hönnun

Hefðbundin nálgun

- Fundir til að skilgreina kröfur og hönnun
- Tekið saman í "Backlog"
- Vinna við POC getur tekið daga eða vikur (ef við á)
- Oft takmarkað samhengi milli teyma

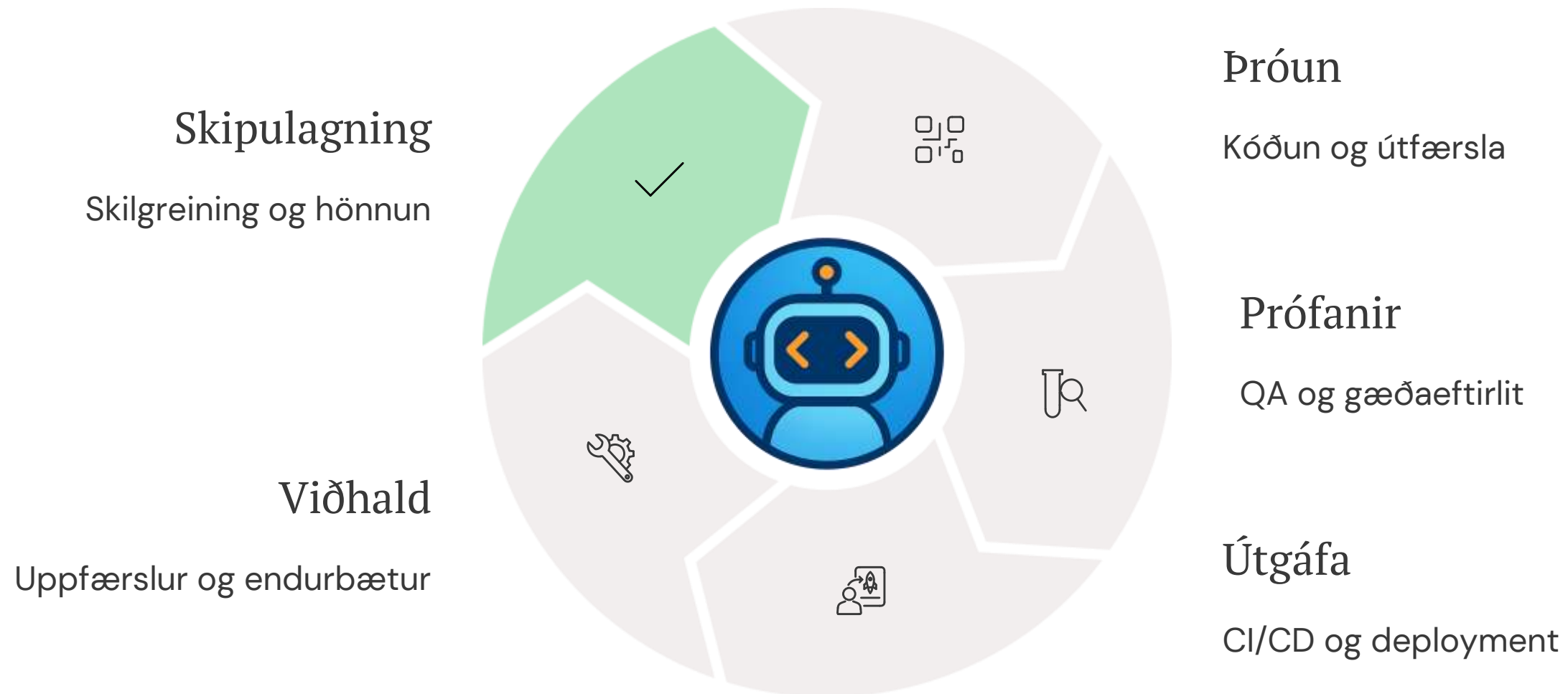
Með AI-stuðningi

- Fundir til að skilgreina kröfur og hönnun
- Slípun með AI, og sent í Ticketing kerfi með MCPs
- POC getur verið tilbúin á mínútum eða klukkustundum
- Betti skilningur á öllum hlutum kóðasafnsins með ítarlegu Context
- Styttri sprettir eða fleiri features





Lífsferill hugbúnaðarþróunar - SDLC





AI í kóðun og þróun



Hefðbundin nálgun

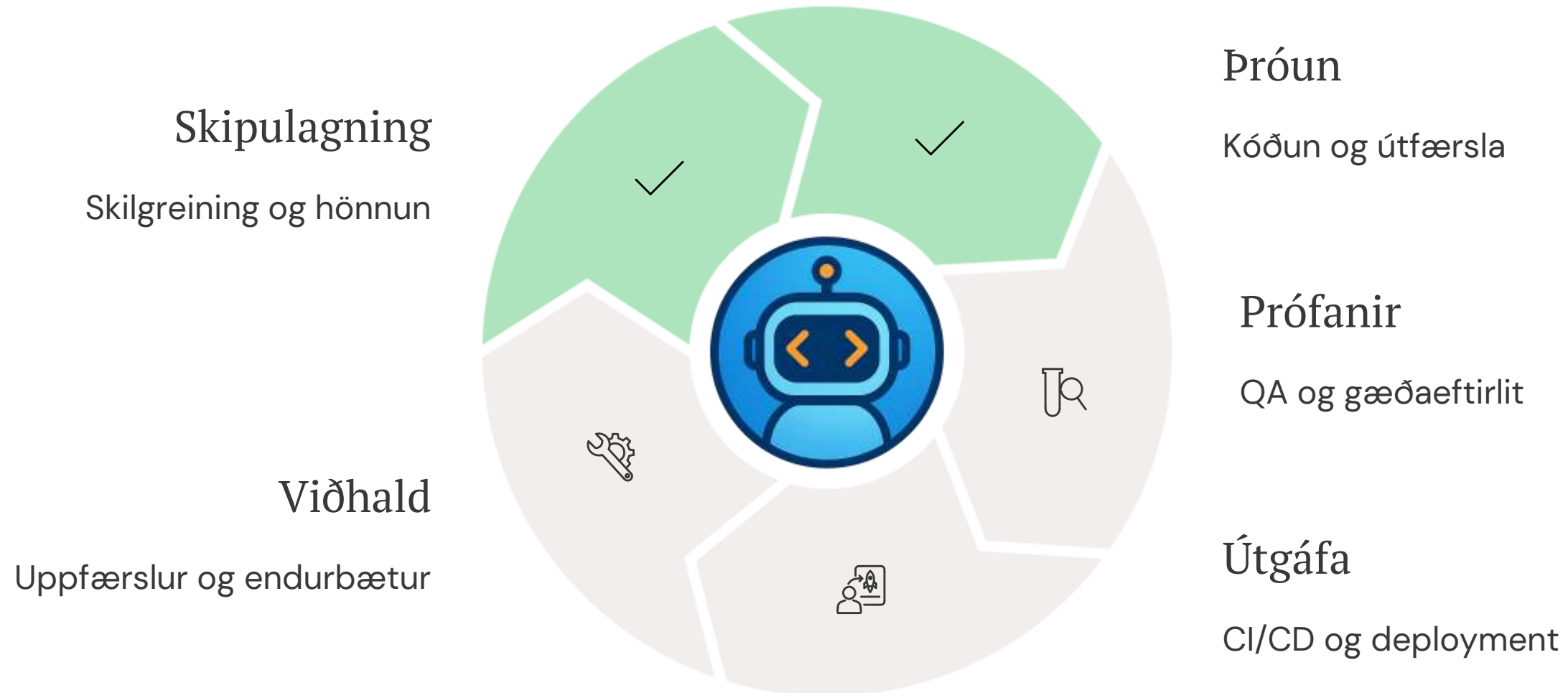
- Þróunaraðilar vinna á afmörkuðum eiginleikum, oft einir eða í litlum teyllum.
- Samskipti um samhengi geta verið tímafrek, sérstaklega milli mismunandi teyma.
- Forðgangsröðun eiginleika byggist á mönnum og getur verið erfitt að sjá heildar ROI.

Með AI-stuðningi

- AI aðstoðar við kóðagerð, gefur tillögur og hjálpar við villuleit í rauntíma.
- Betri teymisvinna þar sem AI skilur og auðveldar að viðhalda samhengi milli teyma.
- AI aðstoðar við forgangsraða sjálfkrafa eiginleikum sem skila hæstu arðsemi (ROI).
- Styttir þróunartíma, fækkar villum og eykur framleiðni teyma.



Lífsferill hugbúnaðarþróunar - SDLC





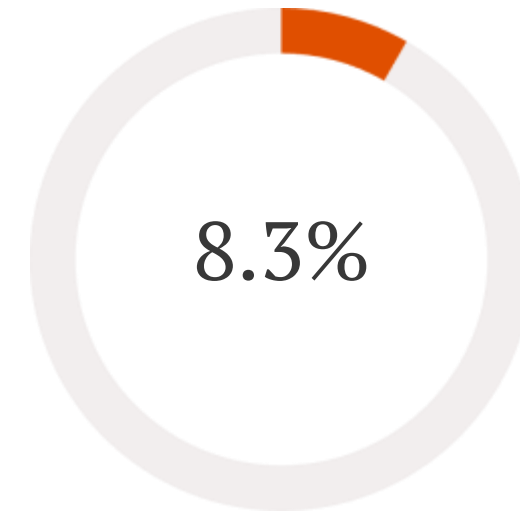
AI í gæðaprófunum og QA

Hefðbundin nálgun

- Handvirkar eða óstöðugar prófanir
- Lítil test coverage
- Prófunargerð oft séð sem leiðinlegt verkefni
- Hætta á regression í eldri kóða
- E2E prófanir oft sleppt

Með AI-stuðningi

- Sjálfvirk prófunargerð fyrir allar tegundir
- Lagfærir óstöðugar prófanir sjálfkrafa
- Eykur test coverage verulega
- Gerir kleift að búa til E2E prófanir sem venjulega eru sleppt
- Minnkar regression áhættu (með context)



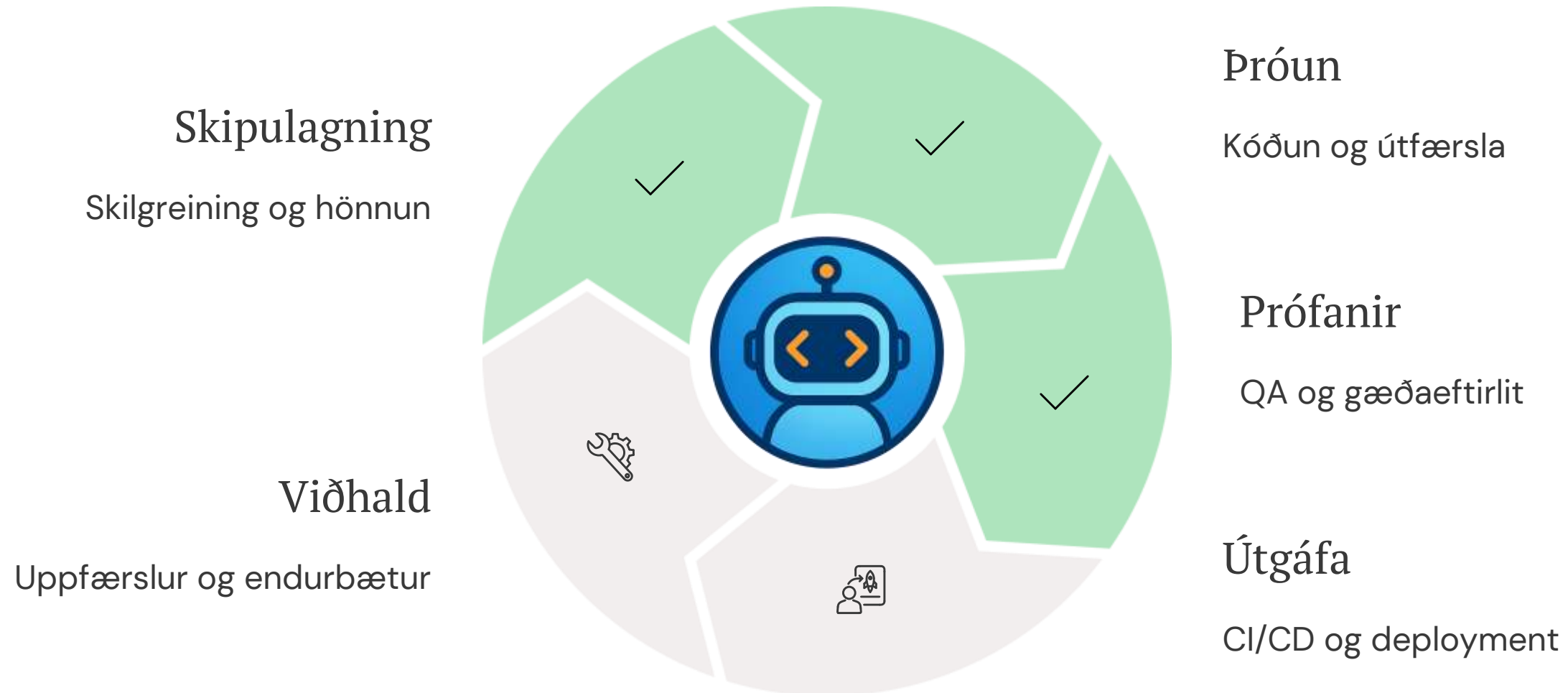
betra success rate í flaky prófunum

Prófunarkeyrslur bötнуðu um **8.3%** þegar gervigreindarkerfi voru notuð við að finna ástæðu.

[Rannsókn frá arxiv.org](https://arxiv.org)



Lífsferill hugbúnaðarþróunar - SDLC





AI í útgáfuferlum



Hefðbundin nálgun

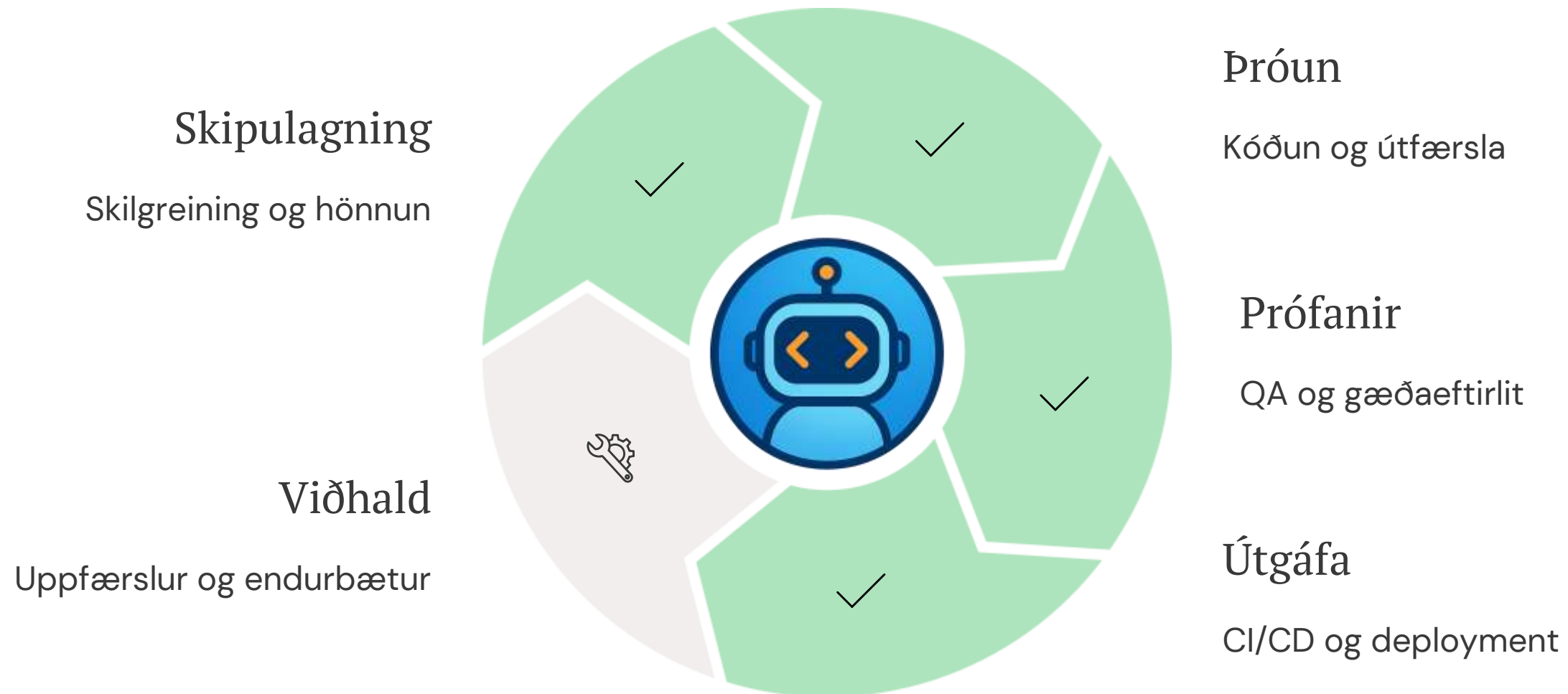
- Handvirkar kóðaskoðanir og gæðaeftirlit.
- Takmörkuð öryggisskönnun, oft sem sérstakt skref.
- Viðbragðstími vegna villna getur verið langur.
- Ferlar byggja á forritunarskrifum og handvirkum aðgerðum.

Með AI-stuðningi

- AI agents skanna kóða fyrir gæði, villur og öryggisholur í rauntíma.
- Fyrirbyggjandi viðvörðun um hugsanleg vandamál í deploymenti.
- Hraðar leiðréttingartillögur og sjálfvirk lagfæring minni háttar vandamála.



Lífsferill hugbúnaðarþróunar - SDLC





AI í viðhaldi: Uppfærslur og endurbætur

Hefðbundin nálgun

- Flestir villur koma fram eftir útgáfu og valda (tímabundinni) tækniskuld.
- Hætta á regression eykst með vaxandi flækjustigi eldri kóða.
- Teymi breytast og óskráð þekkingu um kóða hverfur, skjölun er oft úrelt.
- Handvirk villuleit getur tekið klukkustundir eða daga fyrir hverja villu.
- Uppfærsla libraries og ósjálfstæðra þátta er áhættusöm og krefst endurprófana.
- Framleiðsluvöktun skilar miklum gögnum en greining er tímafrek og oft afturvirk.

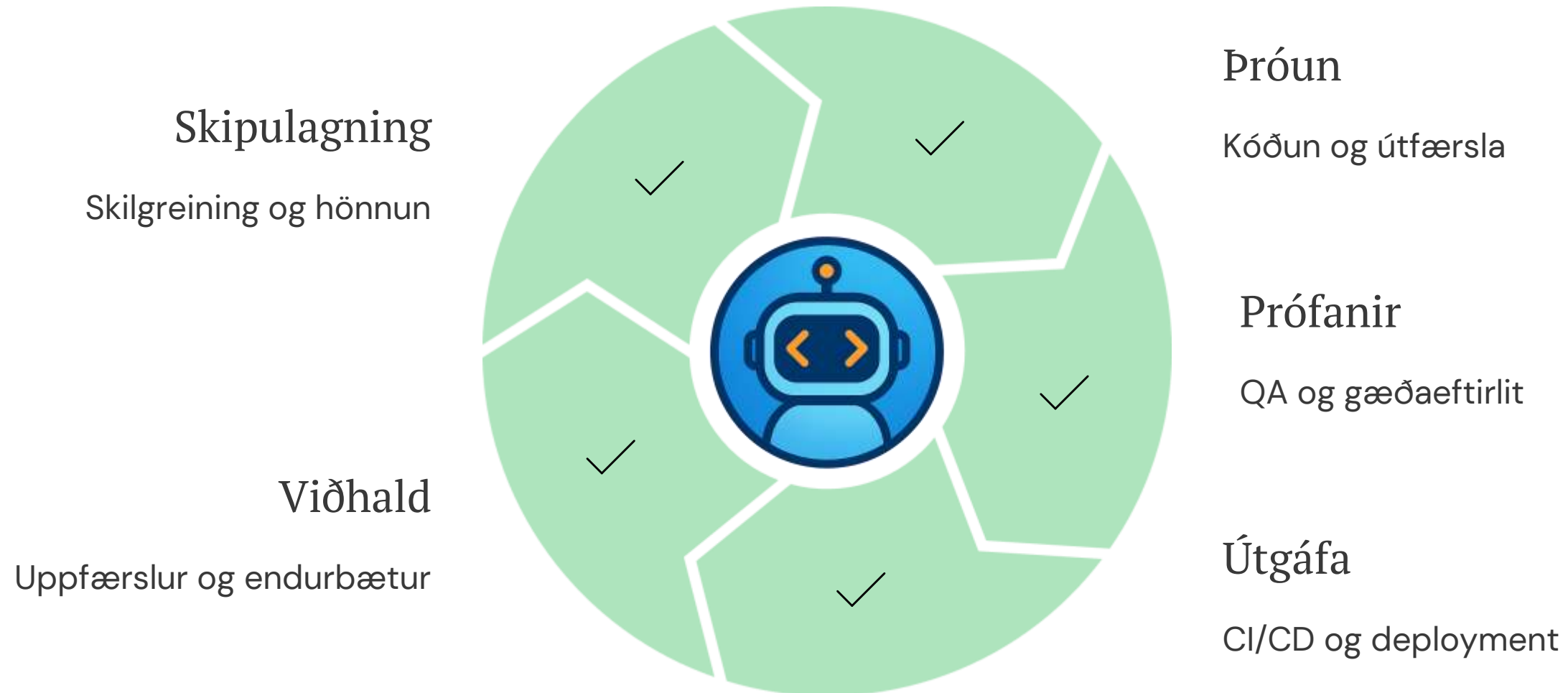
Með AI-stuðningi

- AI-aðstoð við villugreiningu og RCA með sjálfvirkum greiningum á loggum og metrics.
- AI býr til lagfæringar og uppfærslur beint út frá villuskýrslum eða prófunarbilunum.
- Vöktun með gervigreind og fyrirbyggjandi viðhald spáir fyrir um kerfisbilanir áður en þær eiga sér stað.
- AI býr til skjölun sem heldur kóðaskilningi lifandi og breytir "óskráðri þekkingu" í aðgengilegan þekkingargrunn.





Lífsferill hugbúnaðarþróunar - SDLC





Dæmi: Noktun á Cursor og Claude Code



Context is King

Með **Cursor Rules** og **Context Documentation** er auðvelt að tryggja nákvæmt samhengi í kóðagrunni og tryggja réttari tillögur og lausnir.



Sjálfvirkni með agentum

Cursor Custom Commands, Skills, , Subagents og Hooks sjálfvirknivæða verkefni og prófunargerð, stytta þróunartíma og bæta gæði.



Ferlastjórnun

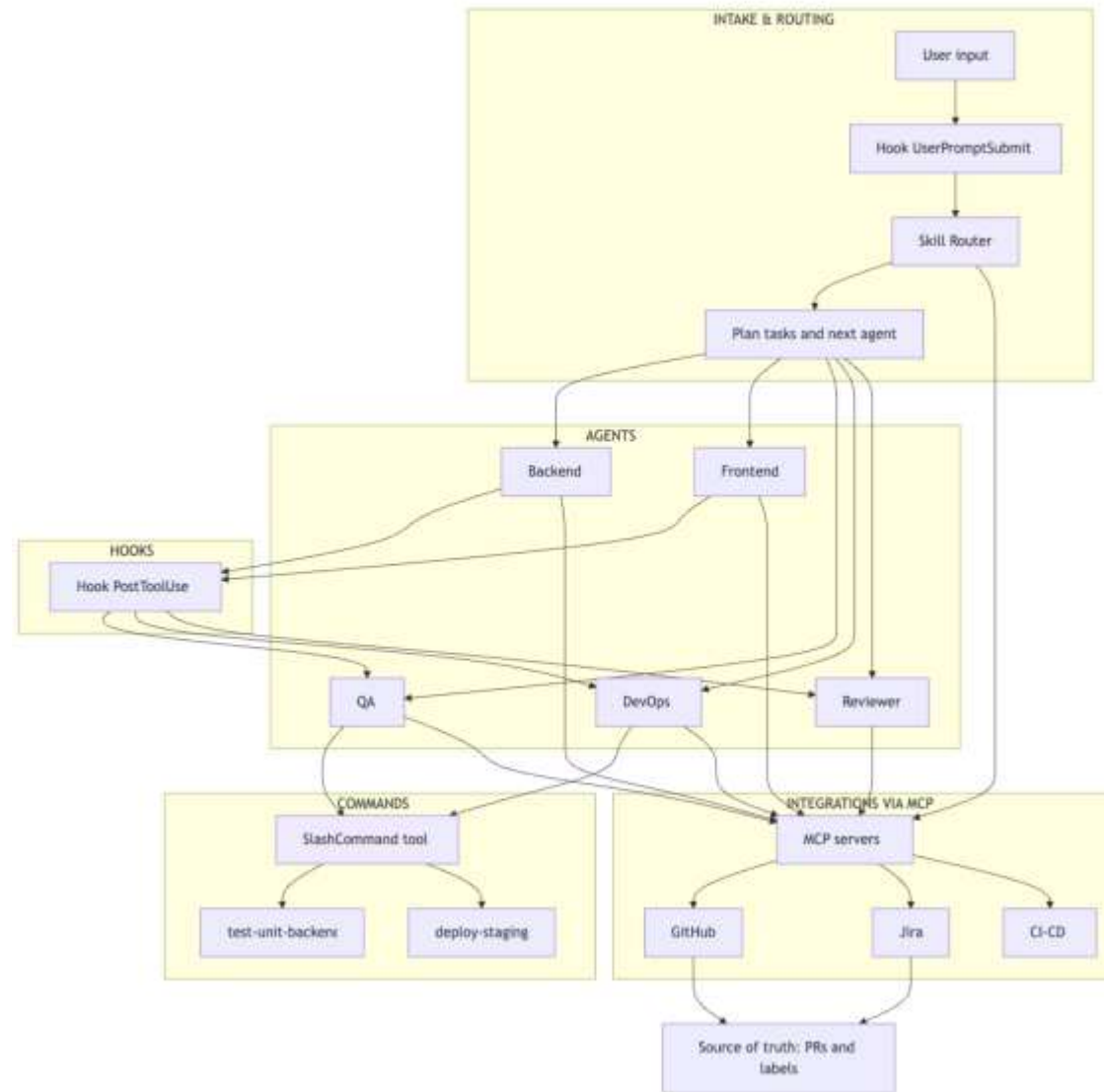
Agent Orchestration gerir kleift að innleiða AI agenta í öll hlutverk SDLC og flýta öllum skrefum líttíma þróunar.

*á við sérstaklega fyrir Cursor, en önnur töl bjóða upp á svipaða uppsetningu og ferla.



Dæmi um feril með Cursor og Claude Code

- Hvert repo er með sömu uppsetningu
- Cursor IDE notað en Claude Code (CLI tól) notað fyrir vinnuna.
- Innleitt í skrefum
- Hver "subagent" prófaður í X tíma
- Skills búin til miðað við tæknistakk og dev preferences
- Hooks löguð miðað við þróunarferla viðkomandi fyrirtækis



- Agents innleiddir í CI/CD sérstaklega
- Notun á Git Worktrees til að "assigna" tösk á agenta
- MCP notaðir til að sækja "Source of truth", Pusha á Git repo, uppfæra task list etc.
- Meta agent fyrir Skill, hook, command gerð
- Viðmótshönnun brainstorming



Vegvísir: Innleiðing AI í SDLC



Auðveldir sigrar

- Byrja með AI kóðaaðstoðara (Copilot, Cursor o.s.frv.)
- Bæta við context skráum í núverandi repos
- Sjálfvirk commit message myndun



Næstu skref

- Context engineering fyrir öll repos
- AI-mynduð unit og integration tests
- Innleiða AI agents í CI pipeline



Langtíamarkmið

- Fullkomið samþætt AI í alla SDLC áfanga
- Styttu sprints miðað við nýtt fyrirkomulag
- Sjálfstæðir AI agents fyrir kóðun, QA og deployment

Best er að byrja á einföldum atriðum og fá reynslu á hvaða atriði gervigreindin getur aðstoðað með.



Takk fyrir!