

Sådan bidrager du til et open source projekt

bidrag indflydelse Guide

Her er din guide for, hvordan du kan bidrage til et open source projekt. Guiden kommer fra [Open Source Guides](#), og OS2 har sidenhen oversat den til dansk med nogle ændringer hist og her. Guiden er lavet til begyndere såvel som veteraner udi open source.

INDHOLDSFORTEGNELSE

[Hvorfor dog bidrage til open source?](#)

[Hvad betyder det at bidrage?](#)

[Anatomien af et open source projekt](#)

[Find et projekt at bidrage til](#)

[En tjekliste før du bidrager](#)

[Klar til at bidrage?](#)

[Hvad sker der efter du har bidraget?](#)

Hvorfor dog bidrage til open source?

Der er mange grunde til at bidrage til open source. For open source kan være en givende måde at lære, undervise og opbygge evner inden for stort set alt. Her er en liste over nogle af grundene til at bidrage til open source:

- Styrke eksisterende evner
 - F.eks. kodning, design af brugergrænseflade, grafisk design, at skrive og organisere.
- Møde mennesker med ens interesser
 - Open source projekter med venlige mennesker kan bygge stærke relationer.
- Undervisning af ens selv og andre
 - Når du arbejder med andre, vil du uden tvivl komme til tidspunkter, hvor du må bede om hjælp eller hvor andre beder dig om hjælp.

Ejer:
Godkender:
Revisionsdato:

Version:
Godkendelsesdato:

- Dyrke sit ry (og sin karriere)
 - Da al open source er offentligt, er open source en gratis platform til at vise, hvad du kan.
- Styrk dine ledelseskompetencer
 - Open source metoden giver mulighed for at øve ledelseskompetencer med håndtering af konflikter, organisering af mennesker og prioritering af arbejdsbyrden.
- Det er empowering at skabe en forskel
 - Det kan være en lille ting, der ikke fungerer på en website, og som du ændrede. Selv den lille ændring er med til i sig selv at skabe følelsen af at have indflydelse og autonomi i eget liv.

Hvad betyder det at bidrage?

Det er nemt at tænke, at alt i open source handler om kode. Det behøver det ikke. Der er andre elementer af projektet, som nemt bliver overset. Så hvad kan man bidrage med?

- Kodning (ja); Ret fejl, skab nye features og gennemse andres kode
- Skrift; Lav dokumentation, nyhedsbrev og/eller tutorials for projektet
- Design; Gå i krig med layout og brugervenlighed
- Organisering; Ryd op i projektets issues, hold det pænt og stil opklarende spørgsmål
- Events; Afholdelse af workshops, sammenkomster og konferencer

Og husk open source behøver ikke være software, selvom det ofte vil være hos OS2.

Anatomien af et open source projekt

Selvom alle open source fællesskaber er forskellige, er der stadig visse fællestræk. I et typisk open source fællesskab er der følgende roller:

Author: Personen, personerne eller organisationen der skabte projektet

Owner: Personen eller personerne der har administrativt ejerskab over organisationen/repository'et (det behøver ikke at være den samme som forfatteren)

Maintainers: Bidragsgivere der er ansvarlige for at drive visionen og lede de organisatoriske aspekter af projektet.

Contributors: Alle der bidrager med noget til projektet

Community Members: Folk der bruger projektet og som måske er aktive i samtaler og kommer med deres holdning.

Ejer: Version:
Godkender: Godkendelsesdato:
Revisionsdato:

Med dokumentationen kommer en række udtryk, som er værd at kende, inden man begiver sig ud i et open source projekt:

LICENSE: Per definition skal alle open source projekter have en open source licens – hos OS2 er det MPL og CC-BY-SA.

README: Instruktionsmanualen der velkommer nye til fællesskabet. Det forklarer, hvorfor projektet er brugbart og hvordan man kommer i gang.

CONTRIBUTING: Mens READMEs hjælper til at forstå, hvordan projektet skal bruges, så forklarer CONTRIBUTING dokumenter, hvordan man bidrager til projektet. Det forklarer, hvilken slags bidrag er nødvendige og hvordan det at bidrage foregår. Men ikke alle projekter har CONTRIBUTING dokumenter.

CODE_OF_CONDUCT: Et adfærdskodeks sætter reglerne for opførsel og er med til at sikre et behageligt miljø. Ikke alle projekter har en CODE_OF_CONDUCT file. Men OS2 har et fælles [code of conduct](#).

Anden dokumentation: Der kan være yderligere dokumentation, som f.eks. tutorials, [governance politikker](#) osv.

For at få et fuldt billede af projektet er det en god idé at gå i arkiverne. Det vil give dig et godt billede af, hvordan fællesskabet tænker og virker.

Issue tracker: Hvor folk diskuterer noget relateret til projektet

Pull request: Der hvor folk diskuterer og gennemser de ændringer, der er i gang

Diskussionsfora/mailling lists: Nogle projekter bruger kun issue tracker til al samtale, men andre bruger flere/andre kanaler.

Synchronous chat channel: I nogle projekter bliver der brugt chat kanaler for almindelig samtale, samarbejder og hurtige udvekslinger.

Find et projekt at bidrage til

OS2 har flere produkter og projekter, som du kan bidrage til. Du finder backloggene på [OS2s Jira](#). Du skal have en brugerkonto for at kunne tilgå projekternes backlog, men det er ikke noget problem at få. OS2s Jira bruges til at styre igangværende udviklingsprojekter og udviklingsønsker. Du kan finde mere om [Jira på OS2viden](#). [OS2s Code of Conduct](#) er gældende på Jira.

Ejer:

Version:

Godkender:

Godkendelsesdato:

Revisionsdato:

Der er mange muligheder i OS2s portefølje, derfor er det smart at starte med det, der er tæt på dig. Det, du bruger eller har planer om at bruge.

Når du tager en tur igennem det projekt, du har valgt, så kig efter hvad der kunne være bedre og hvad der slet ikke virker. Open source er ikke en eksklusiv klub, men et begreb for at se verdens problemer med et problemløsningsperspektiv.

Det kunne f.eks. være at du gennemlæste en README og fandt et dødt link eller en stavefejl. Det kunne også være, at du er ny bruger og fandt noget, som du synes burde være i dokumentationen. I stedet for at ignorere det eller spørge en anden, om de kan ordne det, kan du engagere dig selv i at ændre eller rette op på fejlen. Det er det, open source handler om.

En tjekliste før du bidrager

Denne tjekliste hjælper dig med at se, om projektet er egnet til og modnet nok til at modtage dine bidrag. Se på licens, aktivitet, issues, pull requests og fællesskabs kultur.

- Har det en open source licens? Det kan du typisk se i en fil kaldet LICENSE i the root of the repository.
- Hvornår var den sidste forpligtende aktivitet på master branch?
- Hvor mange bidragsydere har projektet?
- Hvor ofte bidrager folk?
- Hvor mange åbne issues har projektet?
- Besvarer maintainers hurtigt på åbne issues?
- Er der aktiv diskussion på åbne issues?
- Er der nye issues?
- Bliver issues lukkede?
- Hvor mange pull request ser der?
- Besvarer maintainers hurtigt på åbne pull requests?
- Er der nye pull requests?
- Har være været et merge af pull requests fornyelig?
- Er maintainers hjælpsomme i deres svar til issues?
- Er folk venlige i issues diskussioner, diskussionsfora og chat?
- Bliver pull requests gennemset?
- Takker maintainers folk for deres bidrag?

Ejer:
Godkender:
Revisionsdato:

Version:
Godkendelsesdato:

Klar til at bidrage?

Hvordan får du dit bidrag afsted over stepperne på en konstruktiv facon? Hvis du er ny i det open source fællesskab, som du gerne vil bidrage til, så er det vigtigt at få formuleret sig ordentligt. Så lad os starte med effektiv kommunikation.

Det skal du huske i din kommunikation

Før du åbner et issues eller en pull request eller stiller et spørgsmål i en chat, så tænk på disse punkter:

1. Kontekst

Sørg for at give kontekst for dit spørgsmål eller issue for at få alle med. Forklar hvordan du løb ind i fejlen, og hvis du vil foreslå en ny idé, så forklar hvorfor du tror, at det vil være gavnligt for selve projektet.

2. Læs din lektie

Vis at du har prøvet, selvom du ikke kender alt til projektet. Det indebærer at se efter svar i projektets README, dokumentation, issues (åbne eller lukkede), mailing list eller lav en søgning på nettet.

3. Kort og direkte

Vær kort og direkte i din kommunikation. Mange projekter har tonsvis af anmodninger, så chancen for at nogen hjælper dig stiger, hvis du går lige til pointen.

4. Hold det åbent

På Jira, lav kommentarer frem for emails – medmindre det er følsom information, f.eks. en sikkerhedsbrist. Når du holder kommunikationen åbent, er der flere, som kan lære af det. Det ligger også i forlængelse af open source tanken om gennemsigtighed.

5. Vær tålmodig

Udvis tålmodighed når du bidrager. Alle andre i projektet er også mennesker og heller ikke hundred procent inde i alle kringelkroge af projektet.

6. Respekter fællesskabet

Måske er fællesskabet ikke enig med dig i alle beslutninger, og måske følger de ikke din idé. Du kan altid starte din egen fork eller eget projekt, men opfør dig pænt overfor de andre i projektet. Respektfuldt og høfligt.

7. Sørg for at få konteksten med

Ejer: Version:
Godkender: Godkendelsesdato:
Revisionsdato:

Før du gør noget, så sørg for at få det hele med ved at skimme for at se om din idé er blevet diskuteret andre steder. Skim README, issues (åbne og lukkede), mailing list og chat. Brug et par key words og så er du godt på vej. Du kan kommunikere igennem forskellige kanaler.

- Koordinationsgruppe: Løsningens koordinationsgruppe tager udviklingsønsker op fra alle kommuner i projektet. Find kontaktpersoners email på produktsiden under løsninger på os2.eu
- Issues på Jira er som at starte en samtale eller diskussion
- Pull requests på Jira er som at starte et arbejde – hvis du vil åbne et arbejde som leverandøren skal gøre, må det omkring koordinations-/styregruppe inden.
- LinkedIn dialoggruppe: Åben gruppe for offentlige medlemmer og leverandører

Praksis

1. At åbne et issue

Typisk skal du åbne et issue, for at:

- Rapportere en fejl som du ikke selv kan løse
- Diskutere en idé (f.eks. produktet, visionen eller policy)
- Foreslå en ny feature eller en anden projektidé

Tips til kommunikation omkring issues:

- Hvis du ser et åbent issue, som du vil håndtere, så kommenter på det issues, så der ikke bliver lavet dobbeltarbejde.
- Hvis det er et gammelt issue, så spørg om det er blevet håndteret allerede, før du begynder at arbejde på det.
- Hvis du har åbnet et issue og selv finder svaret, så kommenter for at gøre det klart for andre. Evt. dokumenter.

2. At åbne en pull request

Typisk skal du åbne en pull request i disse situationer:

- Angiv trivielle reparationer (f.eks. stavefejl, døde links, en tydelig fejl).
- Starte arbejdet på et bidrag der er ønsket og har været diskuteret som et issue.

Når du arbejder på en pull request, kan det være godt at markere den som "WIP" i titlen (Work in Progress).

Der er flere måder at bidrage med en pull request:

- Lav en fork i repository og klon det lokalt. Se mere om [hvordan en fork merges med repository](#).
- [Lav en branch til dine rettelser](#).

Ejer:
Godkender:
Revisionsdato:

Version:
Godkendelsesdato:

- Lav referencer til alle relevante issues eller vejledende dokumentation i din pull request (f.eks. "Closes #37.")
- Inkluder screenshots af før og efter hvis dine ændringer er i HTML/CSS. Indsæt billederne i din pull request.
- Test dine ændring, men pas på du ikke ødelægger det eksisterende projekt.
- Husk at bidrage i samme stil som projektet bruger.

Hvad sker der efter, du har bidraget?

Grundlæggende kan der ske en af fire ting: **Du får intet svar** (find et andet projekt – projektet kan være gået i stå, der kan være andre grunde til at du ikke får svar, eller noget helt tredje), **nogen beder dig om at ændre i dit bidrag** (vær imødekommende og besvar – hvis du ikke ved, hvordan det løses, så spørg om hjælp), **dit bidrag bliver ikke accepteret** (hvis du er i tvivl om hvorfor, kan du spørge maintainer for feedback) **eller dit bidrag bliver godkendt** (tillykke!)

Viola, du er nu en open source bidrager: "Open source is made by people like you: one issue, pull request, comment, or high-five at a time." Det står på [Open Source Guides' guide](#).