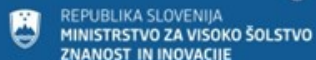


Objavljanje programske opreme

Vid PODPEČAN, Institut "Jožef Stefan"

Ljubljana in online, Centralna tehniška knjižnica Univerze v Ljubljani,
2. usposabljanje za podatkovne strokovnjake, 6. 11. 2025





Kazalo

- Objava programske opreme: kaj, zakaj in kako
- Koraki pri pripravi programske opreme na objavo
 - organizacija, dokumentacija, testiranje, licenciranje, pakiranje, objava
- Praktični primeri in nasveti



Kaj pomeni objaviti programsko opremo?

- to ni isto kot deliti kodo na portalu GitHub!
 - je pa delitev kode z drugimi prvi korak v pravo smer
 - primeri: jezik python, TensorFlow, scikit-learn, BLAST, pyTorch, matplotlib
 - avtorji so delili svojo kodo, ki je kasneje bila tudi objavljena oz. je postala pomembna
- objava zajema
 - dokumentirano in testirano kodo/izvršljiv program (repozitorij ali arhiv)
 - licenco za uporabo
 - možnost citiranja (DOI)
 - najboljše je imeti članek, lahko pa tudi samo arhiviran program
 - javno dostopnost

Zakaj objavljati programsko opremo?

- tudi programska koda je raziskovalni rezultat
 - včasih je dodatek, včasih glavni rezultat
- objava zagotavlja vidnost, transparentnost, ponovljivost
- objava poveča vpliv in uporabnost
- objava spodbuja sodelovanje
- objava programske opreme je skladna
 - s principi odprte znanosti
 - preglednost, dostopnost, ponovljivost, sodelovanje in vključenost v vseh fazah raziskave
 - s FAIR načeli
 - enostavno iskanje, dostop do podatkov, združevanje in ponovna uporaba
- zelo pomembno: objava nam daje zadoščenje o uspešno opravljenem delu!



Od interne skripte do objave

- programska koda se pogosto uporablja le interno
 - ni strukture, ni pravil, ni dokumentacije
 - po določenem času neuporabe koda dejansko postane neuporabna
- objava prinaša dodano vrednost
- primer
 - skripta za vizualizacijo kompleksnih podatkov iz raziskovalnega projekta
 - lahko postane majhna knjižnica, ki jo s pridom uporabljajo po vsem svetu in jo vzdržuje in nadgrajuje par strokovnjakov, ki imajo za to interes
 - če dodamo še objavo, imamo zaključeno celoto
- koraki
 - organizacija, dokumentacija, testiranje, licenca, pakiranje, objava → članek + citati



- <https://github.com/>
- platforma za gostovanje kode in sodelovanje na osnovi sistema Git
- omogoča nadzor različic, “issues”, “pull requests” in avtomatsko testiranje (CI/CD)
- uporaba: deljenje, pregled in prispevanje k raziskovalnim ali odprtokodnim projektom
- omogoča objave (releases) in povezavo z Zenodo za samodejno ustvarjanje DOI
- močna skupnost
- zastonjski paket zadošča za praktično vse potrebe manjših projektov



Zenodo

- <https://zenodo.org/>
- odprta repozitorska storitev, ki jo vzdržuje CERN
- omogoča trajen arhiv raziskovalnih podatkov, programske opreme in publikacij (GitHub je dinamičen, podvržen spremembam)
- samodejno dodeli DOI (Digital Object Identifier) – kodo lahko citiramo
- integracija z GitHub: vsak “release” lahko samodejno ustvari DOI v Zenodu

Organizacija programske kode

project/

├─ README.md

├─ LICENSE

└─ CITATION.cff

├─ src/

├─ tests/

├─ docs/

└─ [setup.py](#)

- pregledna, urejena struktura
- tekstovna datoteka z navodili
- tekstovna datoteka z licenco
- tekstovna datoteka s podatki za citiranje
- glavnina kode v svoji mapi
- dokumentacija v svoji mapi
- konfiguracijske datoteke, nujne za uporabo
 - makefile, [setup.py](#), build.xml, pom.xml, build.gradle, pyproject.toml, ...
- **dobra organizacija olajša razvoj in kasnejšo objavo!**



Dokumentacija programske opreme

- minimalni temelj
 - readme datoteka in primer uporabe
- dobra praksa
 - dokumentiramo sproti in dokumentacijo posodabljammo!
 - dokumentiranje vseh funkcij, modulov
 - pomaga nam lahko dober urejevalnik kode in AI orodja
 - avtomatsko generiranje dokumentacije iz opisov funkcij, modulov
 - Sphinx, MkDocs, ...
 - objavljanje dokumentacije na Readthedocs



Datoteka CITATION.cff

- standardizirana datoteka z metapodatki o citiranju programske opreme
- shrani osnovne informacije
 - naslov in opis programske opreme
 - avtorje in njihove identifikatorje (npr. ORCID)
 - različico, licenco in DOI
 - priporočeni zapis za citiranje
- GitHub samodejno prepozna datoteko in uporabnikom prikaže gumb “Cite this repository”
- omogoča dosledno in strojno berljivo citiranje
- zelo priporočljiva za vse javne in raziskovalne repozitorije programske opreme!

Primer za CITATION.cff

```
cff-version: 1.2.0
title: scikit-learn
type: software
authors:
  - name: "The scikit-learn developers"
message: "If you use scikit-learn in a scientific publication, we would appreciate
citations to the following paper:"
preferred-citation:
  type: article
  title: "Scikit-learn: Machine Learning in Python"
  authors:
    - family-names: "Pedregosa"
      given-names: "Fabian"
      ... (+ še vsi ostali avtorji)
  journal: "Journal of Machine Learning Research"
  volume: 12
  start: 2825
  end: 2830
  year: 2011
  url: "https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html"
```



Testiranje in stalna integracija (CI/CD)

- pogosto zapostavljeno pri akademski kodi!
- ena od bolj zoprnih nalog
 - v pomoč so nam lahko AI orodja
- namen: potrditev, da koda deluje po pričakovanjih
 - preprečevanje napak pri spremembah kode
 - pomoč pri razumevanju
 - povečanje zanesljivosti
- vrste
 - enotski test - posamezne funkcije
 - integracijski test - ali deli sistema pravilno sodelujejo
 - testi primerov uporabe - preverjanje celotnih scenarijev uporabe kode
- CI/CD
 - stalna integracija in stalen “deploy”: testiranje ob vsaki spremembi, če je vse v redu, se objavi nova verzija



Licenciranje programske opreme

- izbira licence je odvisna od različnih faktorjev
 - kaj bomo dovolili uporabnikom
 - če uporabljamo tujo kodo, moramo zagotoviti kompatibilnost licenc
- pogoste licence za raziskovalno programsko opremo
 - preprosta, odprta: MIT, BSD
 - zagotovitev ohranjanja odprtosti kode: GPL, LGPL
 - eksplicitno dovoljenje uporabe patentov: Apache 2.0



Pakiranje programske opreme

- namen: olajšati delo končnim uporabnikom
- pogosto se izvaja avtomatsko
 - GitHub Actions
- oblika je odvisna od vrste programske opreme
- veliko možnosti
 - namestitveni program, izvršljiva datoteka
 - programski paket, npr. za pypi (Python), Anaconda (Python), CRAN (R), npm,...
 - celotno okolje (docker slika)
- primer namestitve python paketa pandas za obdelavo podatkov
 - `pip install pandas` (paket se poišče v repozitoriju pypi in se namesti)



Objava programske opreme

- ko je programska oprema zrela za delitev s svetom
- tri glavne poti
 - programska oprema je dodatek znanstvenemu članku
 - pogosto v akademskem svetu
 - največ vredni citati (visok IF)
 - programska oprema je glavnina članka (t.i. software paper)
 - pogosteje v zadnjem času: JOSS, JMLR MLOSS, SoftwareX
 - tipično nižji IF ali tudi brez IF
 - programska oprema je arhivirana (ima DOI)
 - ZENODO (+ GitHub)
 - mogoče je citiranje, nimamo pa formalne objave (ne šteje pri napredovanjih ipd.)



Objava programske opreme (2)

- primeri slabih praks
 - samo GitHub repozitorij
 - oteženo citiranje, za akademski svet skoraj brezvredni citati
 - samo objava kode na domači spletni strani
 - zelo pogosto v preteklosti, danes redko
 - oteženo citiranje, verzioniranje, sodelovanje
 - patentiranje algoritmov, stroge licence
 - zavira razvoj, uporabo, sodelovanje



Objava programske opreme: primer 1

- situacija
 - imamo uporaben program, zanimiv za širšo javnost, manjši prispevek k znanosti, potrebujemo revijsko objavo z IF za napredovanje, magistrerij, doktorat
- postopek
 - priprava: uredimo kodo, dokumentiramo, testiramo, licenciramo, pakiramo
 - ustvarimo javen GitHub repozitorij, ga dodamo v Zenodo, dobimo DOI
 - napišemo kratek članek in pošljemo v recenzijo v revijo SoftwareX (IF=2.4)
 - če nismo uspešni, izberemo skromnejšo revijo in ponovimo: Software Impacts (IF=1.2)



Objava programske opreme: primer 2

- situacija
 - naredili smo raziskavo, zraven veliko programirali, koda je neločljiv del rezultatov, imamo očitni prispevek k znanosti, želimo objaviti in dobiti citate
- postopek
 - priprava: uredimo kodo, dokumentiramo, testiramo, ...
 - ustvarimo javen GitHub repozitorij
 - napišemo znanstveni članek za ustrezno domensko revijo, programska koda je supplement
 - programske kode recenzenti pogosto niti ne pogledajo...
 - lahko ciljamo na razmeroma visok IF, vendar je poudarek na raziskavi, ne na programu



Objava programske opreme: primer 3

- situacija
 - implementirali smo nov uporaben program v specialnem področju, npr. biologiji, želimo objaviti in dobiti citate
- postopek
 - priprava: uredimo kodo, dokumentiramo, testiramo, ...
 - izberemo primerno pakiranje: spletna aplikacija, samostojni program, ...
 - napišemo članek za revijo, ki omogoča takšne objave
 - npr. Bioinformatics application note, vsakoletna julijska izdaja Nucleid acids research, ...
 - tipično visok IF, članek mora biti kvalitetno napisan, program delujoč in uporaben



Objava programske opreme: primer 4

- situacija
 - program, ki smo ga dolgo razvijali, želimo deliti s svetom, citati nas ne zanimajo, želimo pa si prepoznavnosti kot avtor programa
- postopek
 - priprava: uredimo kodo, dokumentiramo, testiramo, ...
 - posebej pazimo pri izbiri licence, dokumentaciji
 - določimo pravila sodelovanja (code of conduct)
 - ustvarimo javen GitHub repozitorij
 - če je primerno, objavimo paket v ustreznem repozitoriju (pypi, CRAN, npm, ...)
 - aktivno sodelujemo v nadaljnjem razvoju, rešujemo težave uporabnikov
 - smo odprti za ideje in prispevke sodelavcev
 - skrbimo za redne popravke, nadgradnje
 - repozitorij ostane “živ”
- ko je program dovršen in uporaben, lahko napišemo tudi članek za JOSS, če želimo



Objava programske opreme: primer 5

- situacija
 - imamo uporaben program, ki smo ga skrbno razvijali, rabimo revijsko objavo, ne rabimo pa nujno IF
- postopek
 - priprava: uredimo kodo, dokumentiramo, testiramo, ...
 - enako kot primer 4
 - izberemo primerno revijo: JOSS, SoftwareX, F1000Research



Povzetek

- objava je smiseln zaključek razvoja programske opreme
- objava zahteva, da smo pri razvoju sistematični in skrbni
 - ali pa moramo za vse poskrbeti na koncu, kar je veliko težje!
- izberimo pravi način objave
 - revije, ki objavljajo izključno programsko opremo, so v vzponu
 - JOSS se trudi pridobiti JIF, vendar je tudi brez njega ugledna revija
- programska oprema je raziskovalni dosežek, pogosto tudi prispevek k znanosti

Ne pozabi:

Publishing software is publishing your methods

(objava kode je enako pomembna kot objava raziskovalnega članka)

