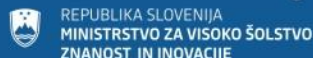


Verzioriranje in uporaba Git, GitHub/GitLab

Katja MEDEN, mag. inf. in zal. štud., Institut Jožef Stefan in Inštitut za
novejšo zgodovino

Ljubljana in online, Centralna tehniška knjižnica Univerze v Ljubljani,
Usposabljanje podatkovnih strokovnjakov, 18. – 21. 11. 2024



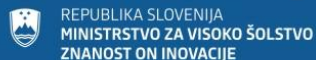


Kazalo vsebine

1. Nadzor različic (version control systems)/Verzioniranje
 - Zakaj nadzor različic?
 - Lokalni, centralizirani in distribuirani sistemi nadzora različic
2. Git – osnovni koncepti
 - Osnovni termini in koncepti
 - Git delotok
3. GitHub in GitLab
 - (Git in) GitHub
 - (GitHub in) GitLab
4. Povzetki in zaključki

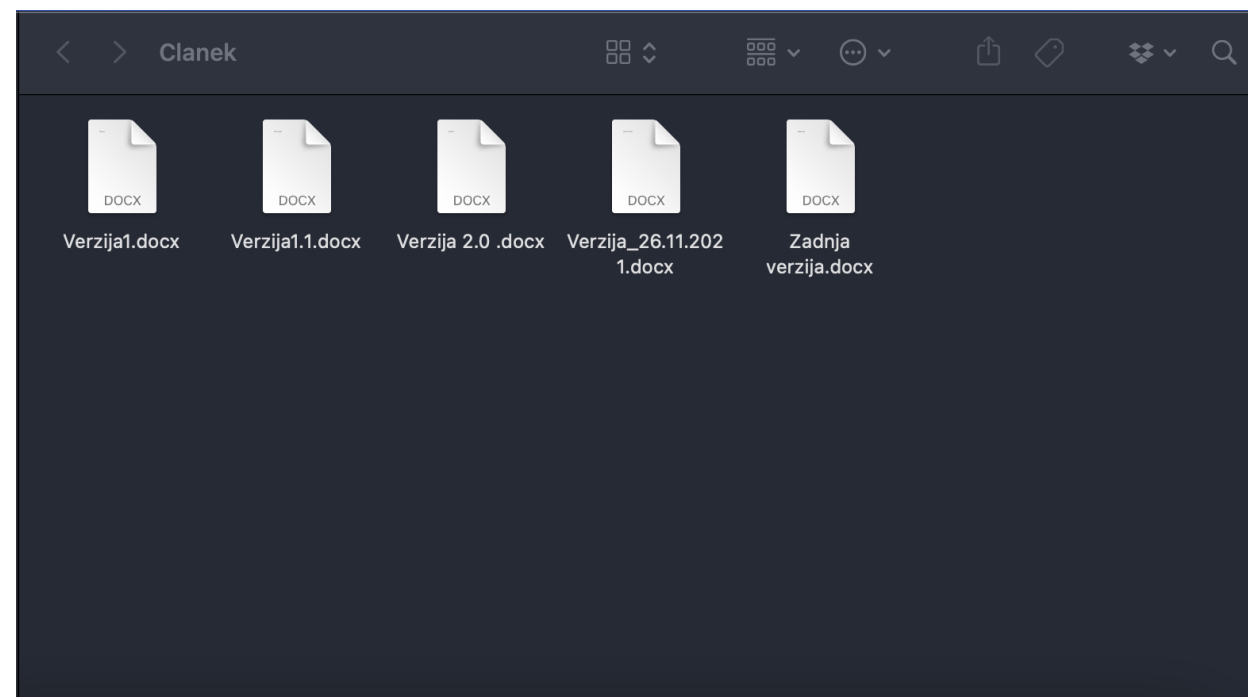
Nadzor različic/Verzioriranje

Kaj je Git?



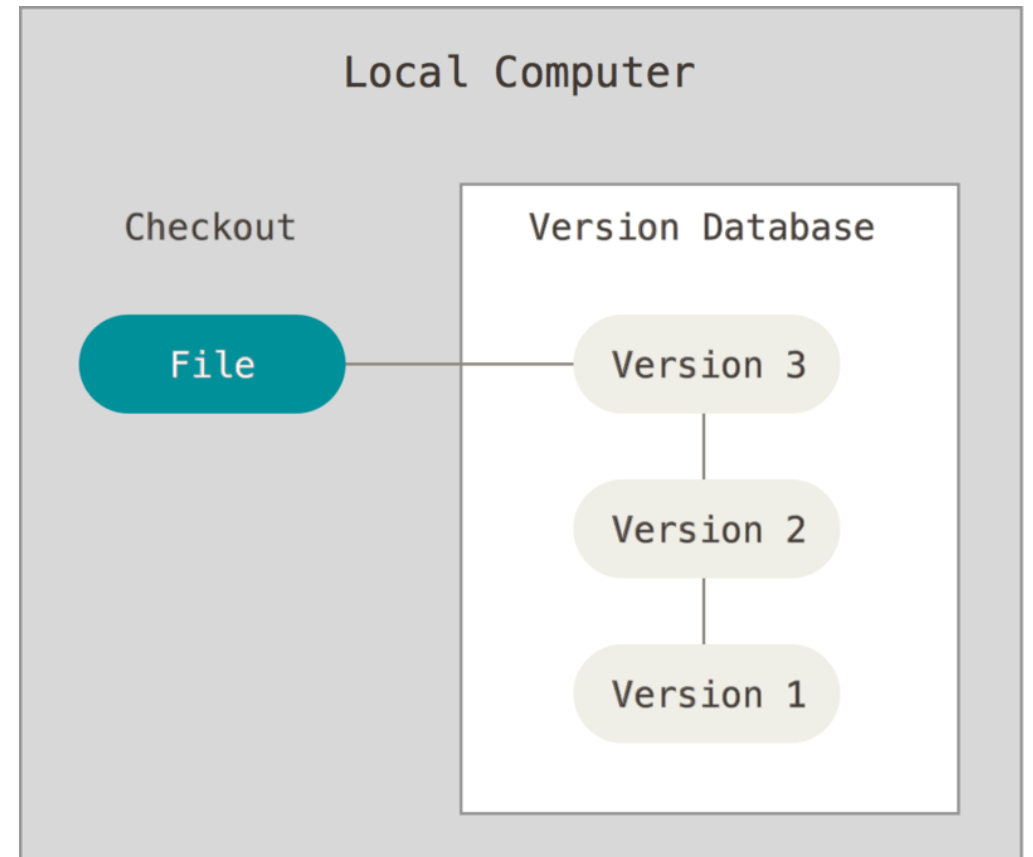
Nadzor različic/Verzioniranje

- Nadzor različic (ang. Version Control, VC): sistem, ki zapisuje spremembe datotek in omogoča sledenje vsaki verziji projektnih datotek...
- Izvorno namenjen nadzoru različic programske kode, vendar primeren tudi za besedile datoteke (.txt, .csv, .tsv, .html, .xml ...)
- Sledi lahko tudi binanim datotekam (npr. Word, Excel, vendar tu ne sledi razlikam v vsebini)
- Vrste VC:
 - Lokalni sistemi nadzora različic
 - Centraliziran sistemi nadzora različic
 - Distribuirani sistemi nadzora različic



Lokalni sistem nadzora različic

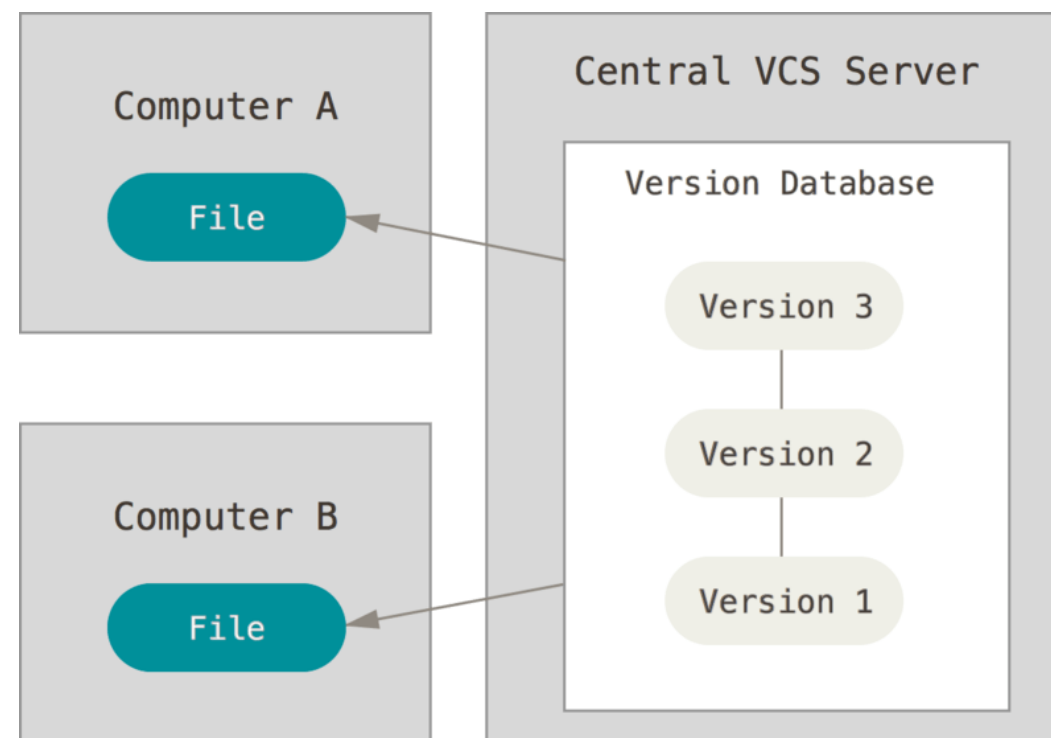
- Zapisovanje in sledenje različicam datotek na lokalni računalnik
- **Prednosti:** avtomatsko sledenje spremembam
- **Slabosti:** Samo en uporabnik, sodelovanje ni mogoče





Centralizirani sistemi nadzora različic

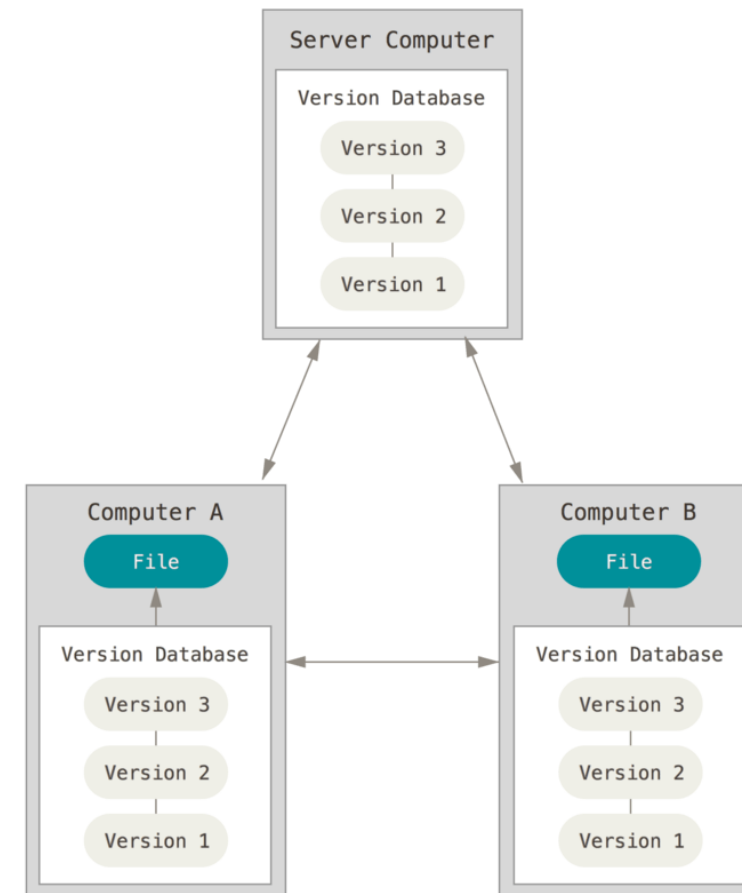
- ang. Centralized Version Control System
- En (centralni) strežnik, ki hrani vse različice datotek in omogoča odjem teh datotek posameznim klientom (uporabnikom).
- **Prednosti:** Omogoča sodelovanje med razvijalci
- **Slabosti:** možnost izgube podatkov v primeru okvare diska na strežniku; nedelujoč strežnik - onemogočeno delo in sodelovanje
- Primer centraliziranega sistema: Subversion



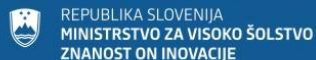


Distribuirani sistemi nadzora različic

- ang. Distributed Version Control System
- Celotna zgodovina sprememb datotek je zrcaljena na lokalnem računalniku vsakega uporabnika
- Prednost: vsaka kopija = varnostna kopija vseh podatkov, omogočeno delo brez povezave (offline)
- Primeri distribuiranega sistema: Git, Mercurial
- Git kot najbolj pribljubljen sistem nadzora različic



Git - Osnovni koncepti

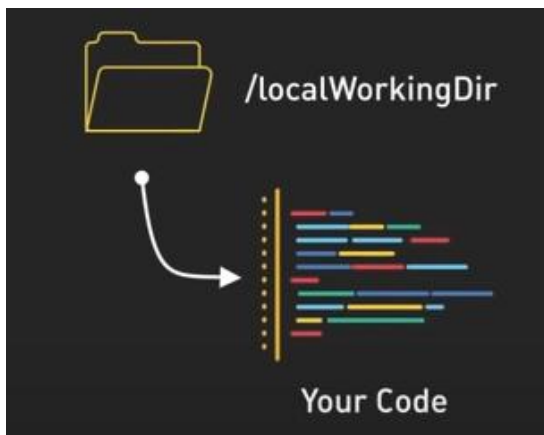
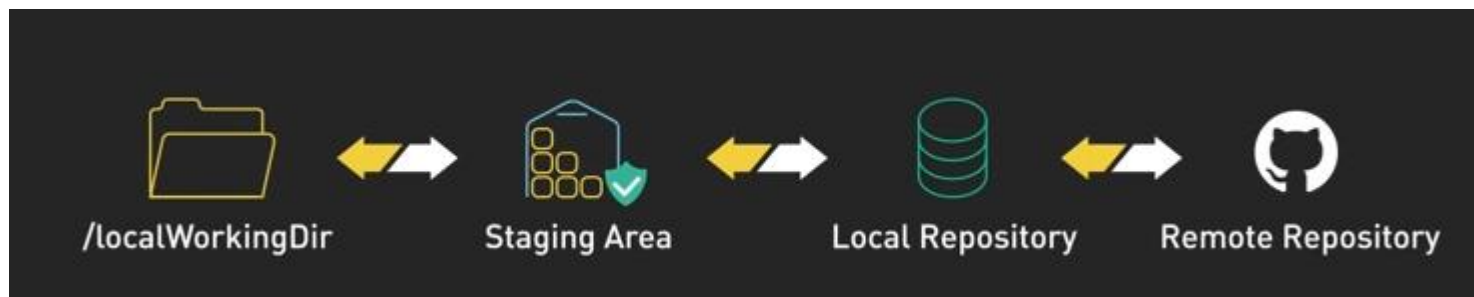


- Najbolj razširjen distribuiran sistem nadzora različic
- Brezplačen, hiter, odprtokoden
- Omogoča:
 - Sledenje spremembam v datotekah/projektu
 - Podpora vejam (branches), npr. “razvoj” in “uporaba”, njihovo združevanje itd.
 - Označevanje konfliktov med različicami
 - Sodelovanje med razvijalci
- Kako lahko uporabljamo Git?
 - Ukazna vrstica (Terminal/CMD/Git Bash)
 - Integracija Git v obstoječih urejevalnikih kode (npr. [Visual Studio Code](#))
 - Namenski urejevalniki (npr. [GitHub Desktop](#))
- Slabosti:
 - Kompleksna arhitektura Gita, težave pri mejnih primerih
 - Napredna uporaba sistema Git zahteva poglobljeno razumevanje njegovega notranjega delovanja

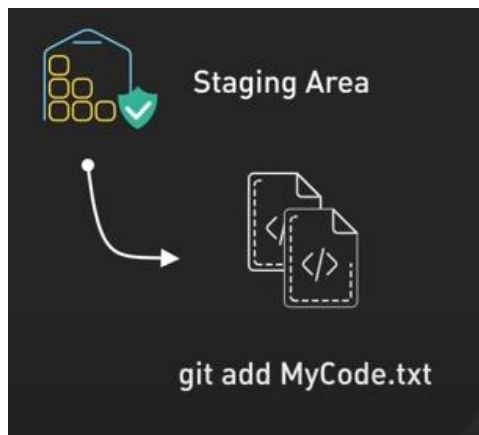
Git – slovarček osnovnih terminov/konceptov

- **Working directory / Delovni direktorij** = Direktorij (mapa) v lokalnem računalniku v katerem urejamo datoteke
- **Local Repository (repo)/ Lokalni repozitorij** = lokalni repozitorij, v kateri so shranjene oddane spremembe
- **Remote repository/Oddaljeni repozitorij** = Repozitorij/Strežnik (npr. GitHub, BitBucket), ki se ga uporablja za deljenje in varnostno kopiranje/hrambo podatkov
- **Git directory/direktorij Git** = V direktoriju Git so shranjeni metapodatki in zbirka podatkov o projektu.
- **Staging area** = začasna faza, ki hrani informacije o tem, kaj bo vključeno v naslednjo oddajo (commit).
- **Commit** = Posnetek trenutnega stanja datotek v projektu
- **Push(ing)** = Prenos sprememb iz lokalnega repozitorija v oddaljeni repozitorij
- **Pull(ing)** = Prenos vsebin iz oddaljenega repozitorija v lokalni repozitorij, s tem posodobimo lokalni repozitorij tako, da se ujema s stanjem oddaljenega repozitorija.

Git - shema



Delovni direktorij (Working Directory): naš lokalni direktorij, znotraj katerega urejamo datoteke



Staging area (index): začasna faza, ki hrani informacije o tem, kaj bo vključeno v naslednjo oddajo (commit).



Lokalni repozitorij: repozitorij, v katerem so shranjeni posnetki trenutnega stanja naših datotek (lokalna naprava)



Oddaljeni repozitorij: Repozitorij, (npr. GitHub/BitBucket), v katerem delimo in hranimo naše podatke

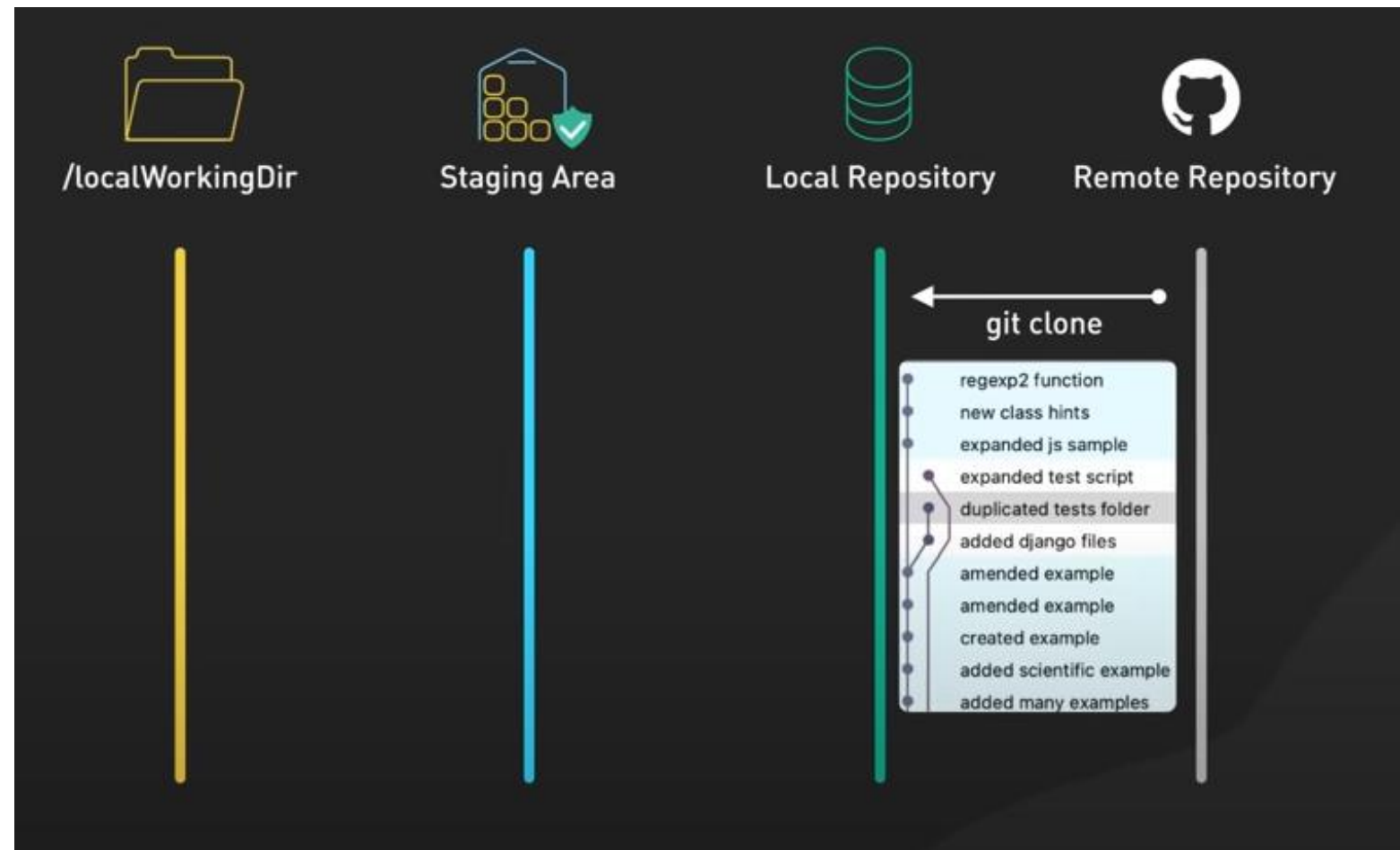
Inštalacija Gita: <https://git-scm.com/downloads>

Začetek:

- Inicializacija novega lokalnega repozitorija, **ali**
- [Kloniranje obstoječega oddaljenega repozitorija](#)

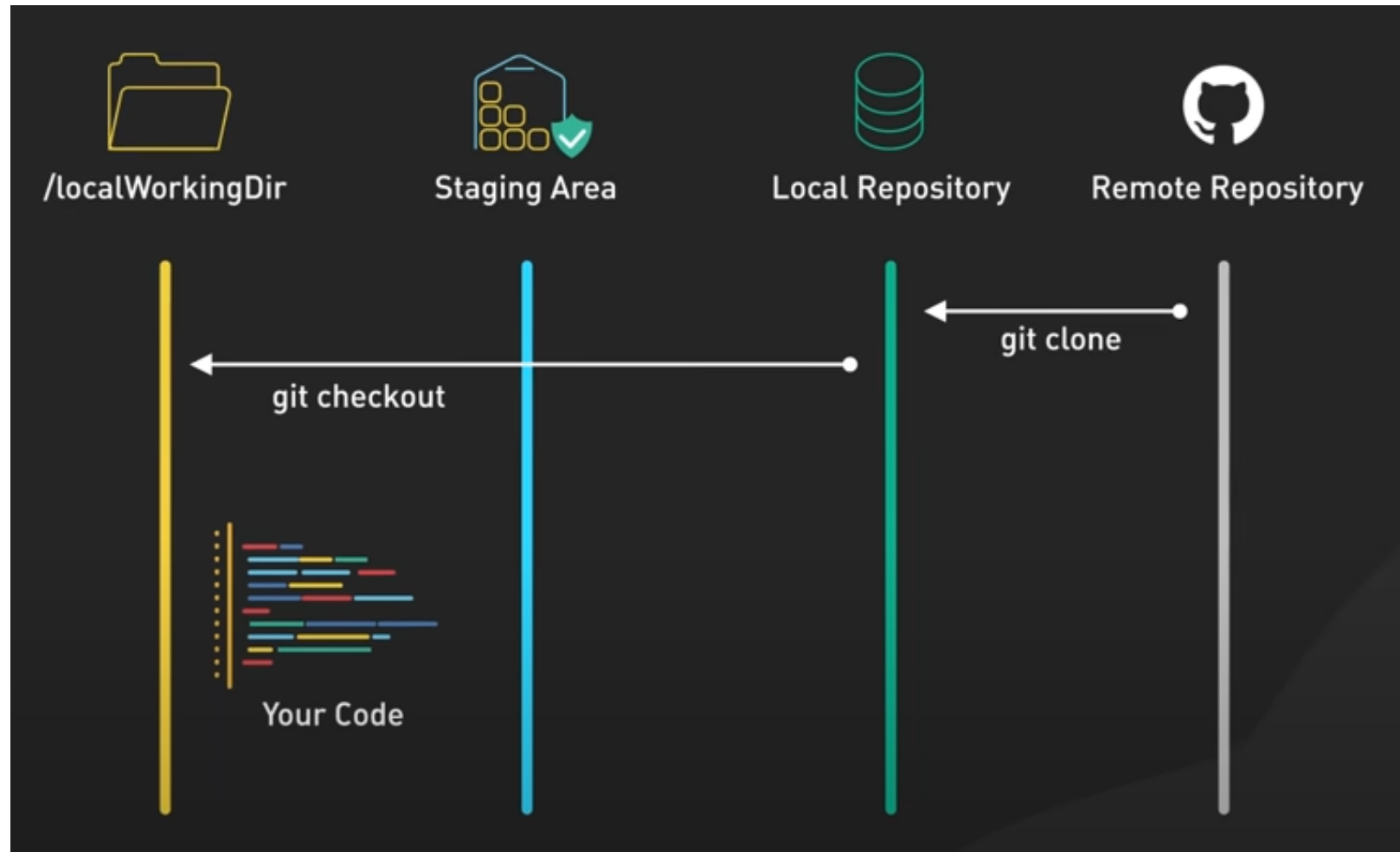
Oddaljeni -> Lokalni repozitorij:

- Prenos/kloniranje obstoječega oddaljenega repozitorija
- Prenos tako podatkov (datotek) kot celotne zgodovine sprememb



Lokalni repozitorij -> Delovni direktorij:

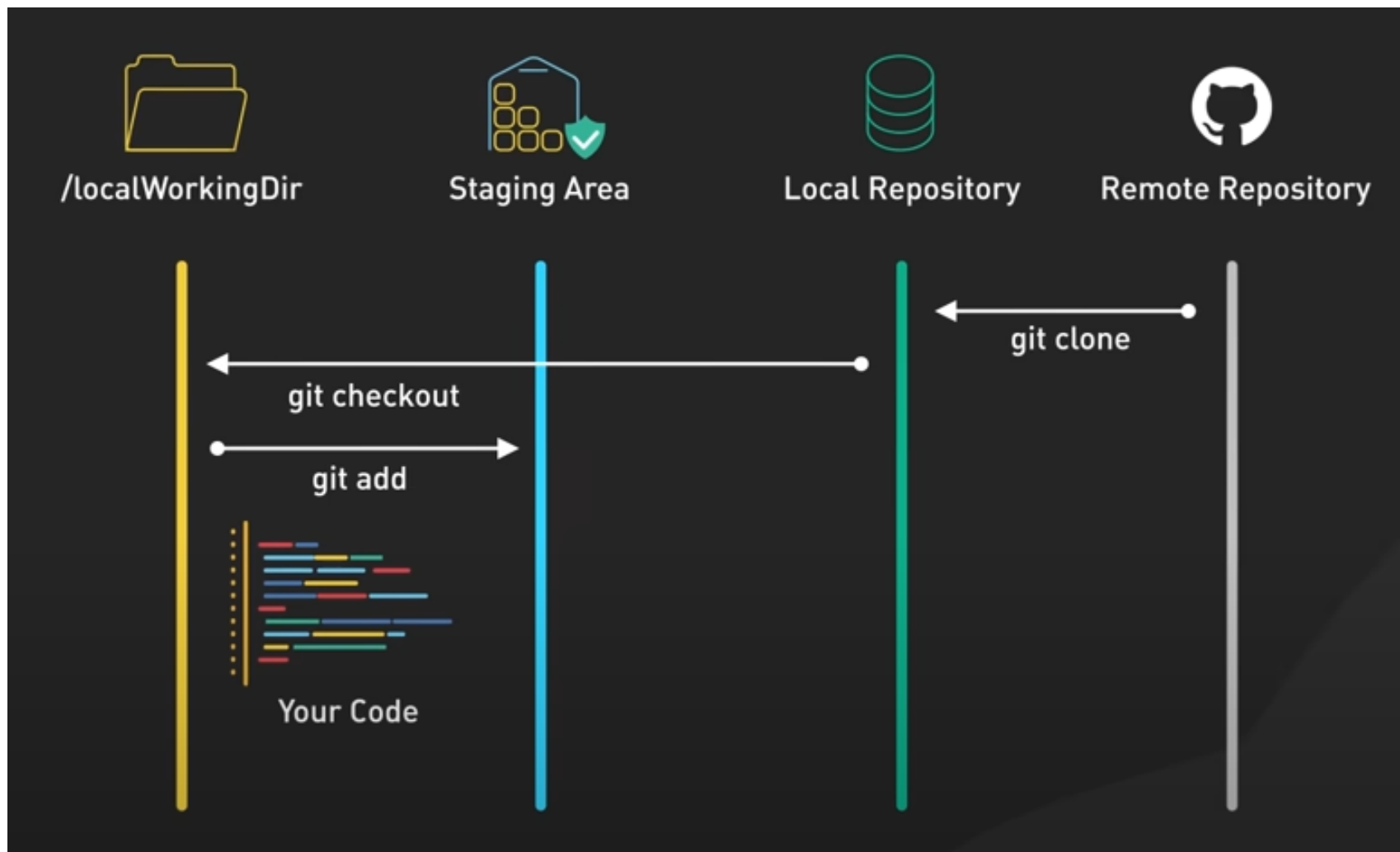
- Spremembe in nadaljnje delo poteka v delovnem direktoriju
- Tu razvijamo svojo kodo/podatke (lokalno delovno okolje)



Delovni direktorij ->

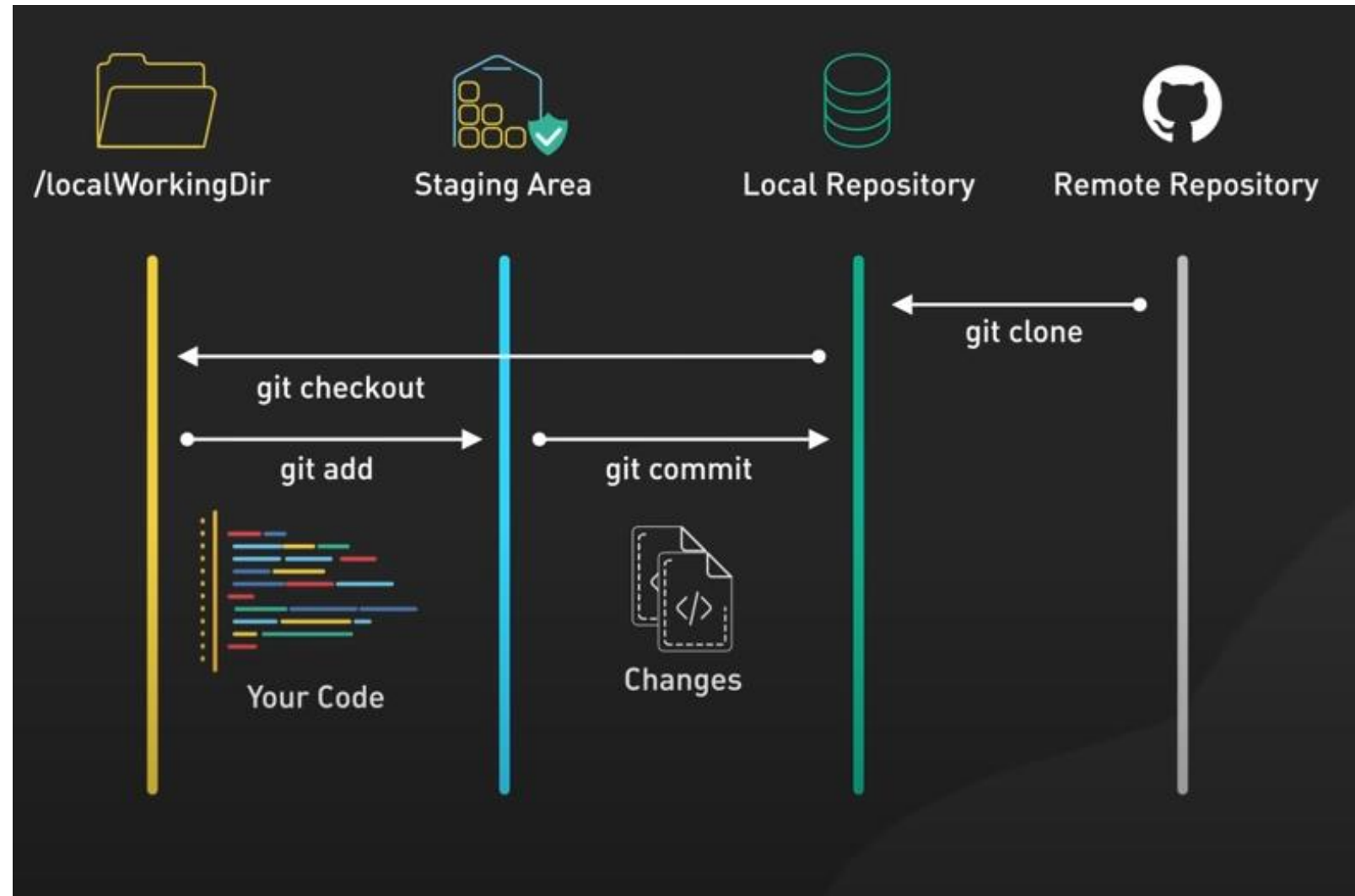
Staging Area:

- Ko zaključimo z delom, nove spremembe začasno dodamo v Staging area
- Tu lahko preverimo/se odločimo, katere datoteke bomo dodali v commit.
- Neke vrste kontrolna točka preden naše spremembe oddamo (commit)



Staging Area -> Lokalni repozitorij:

- Git commit: Spremembe iz Staging area shrani v lokalni repozitorij (snapshot)
- S tem ustvari trajni zapis (commit)
- Commit vsebuje različne (meta)podatke (ID, sporočilo, datum in avtor commita, celotni posnetek sprememb)



```
commit 0dc0d2aa728054192c976e8b3d596f3cd54a2288 (HEAD -> main, origin/main, origin/HEAD)
Author: katjameden <katja.meden@ijs.si>
Date: Mon Oct 28 15:56:14 2024 +0100
```

Add weighted method

```
commit c76fe0d6f20995d35310b19a76b882268d352eae
Author: katjameden <katja.meden@ijs.si>
Date: Fri Oct 25 15:38:34 2024 +0200
```

Calculate basic aggregations and diffs

```
commit 95cb47551d8f4fbcda1dbd92fc4d389e5721800c
Author: katjameden <katja.meden@ijs.si>
Date: Wed Oct 23 16:02:18 2024 +0200
```

Add other aggregations

```
commit 08e4327d0a5dee3ab7ca083e8a1020aa12456ec2
Author: katjameden <katja.meden@ijs.si>
Date: Tue Oct 22 16:46:49 2024 +0200
```

Update data merging and add test workflow

COMMIT

ID

Message

Date/time

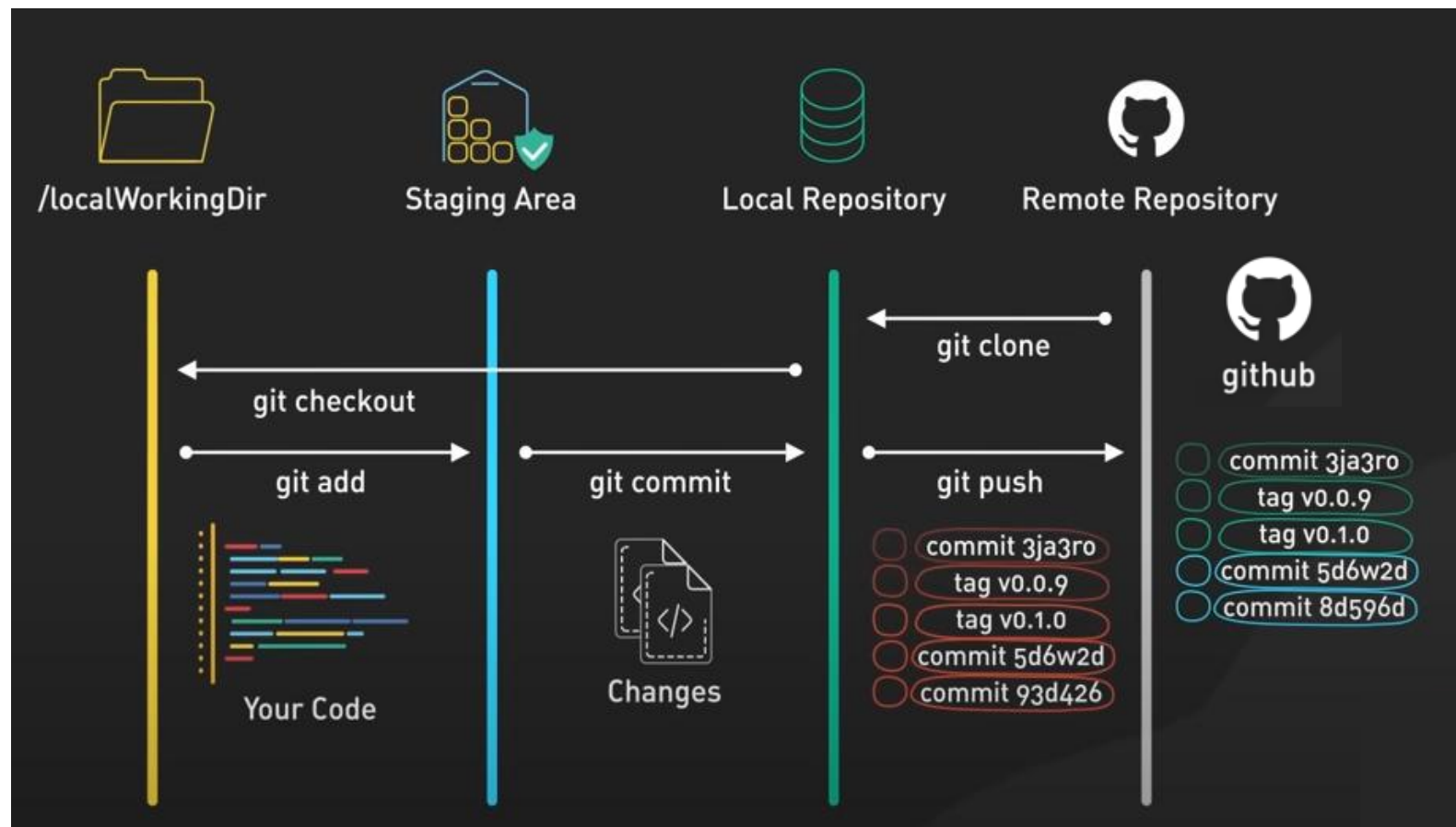
Author

Complete snapshot

Lokalni repozitorij ->

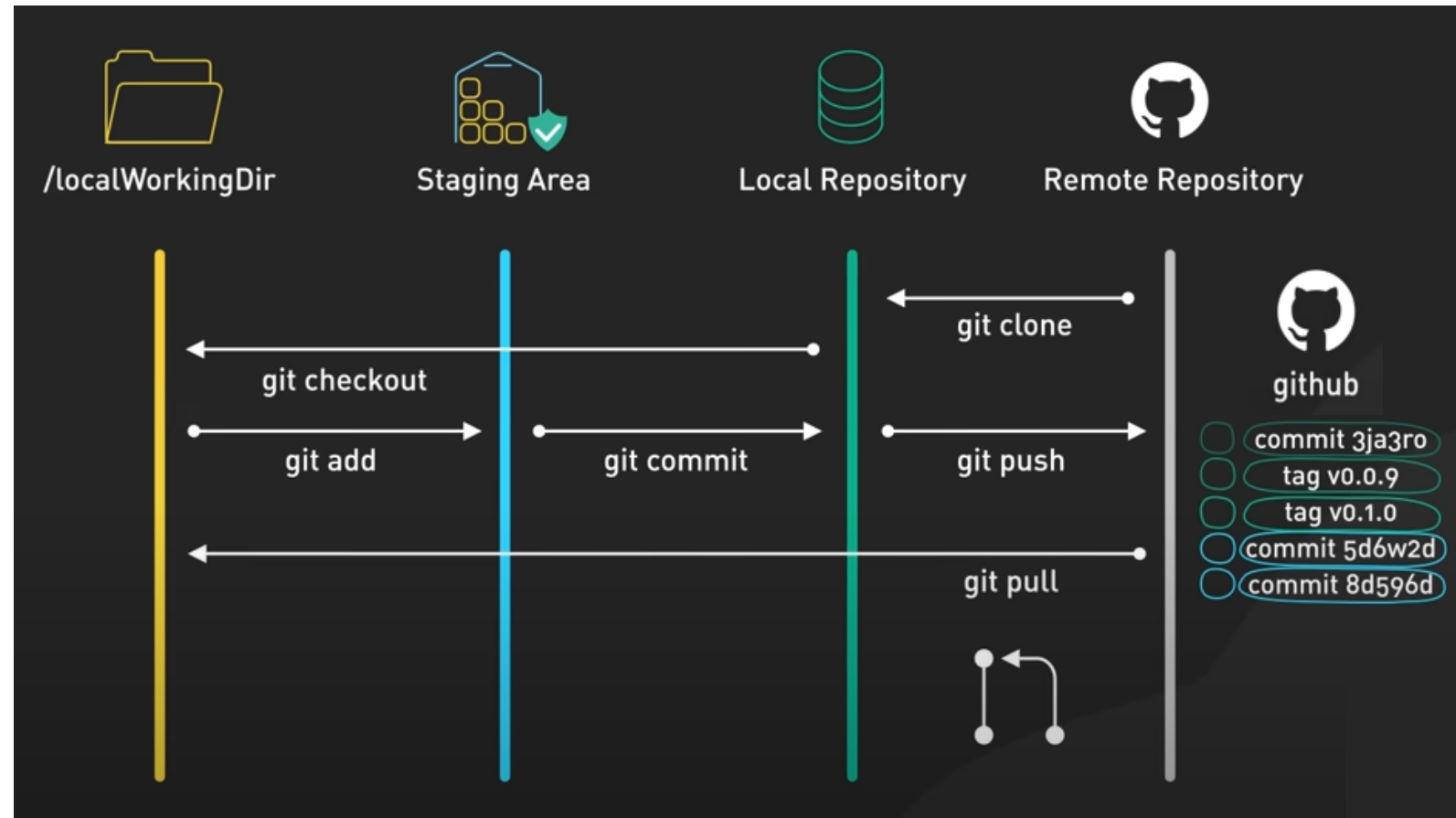
Oddaljeni repozitorij:

- Na tej točki so spremembe shranjene v našem lokalnem repozitoriju/lokalnem računalniku
- Naše spremembe lahko potisnemo (git push) tudi v oddaljeni repozitorij (npr. strežnik, GitHub, BitBucket...)

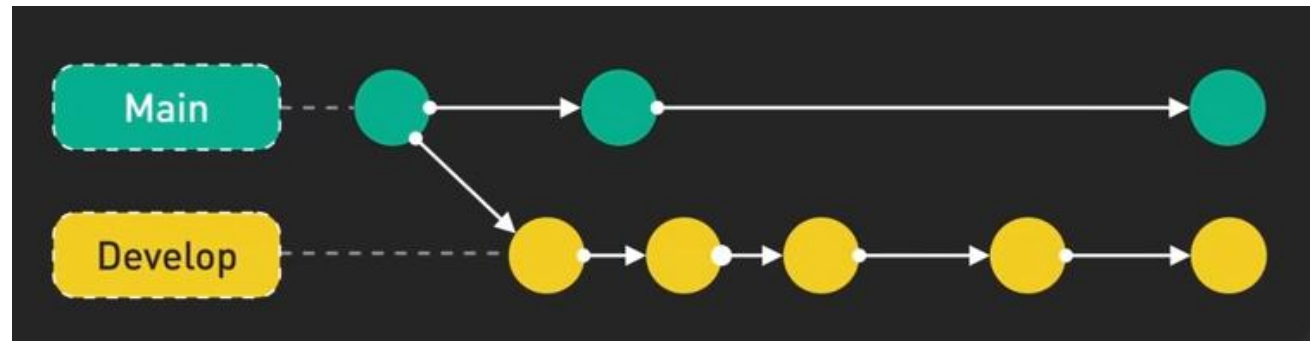


Oddaljeni repozitorij -> Lokalni repozitorij:

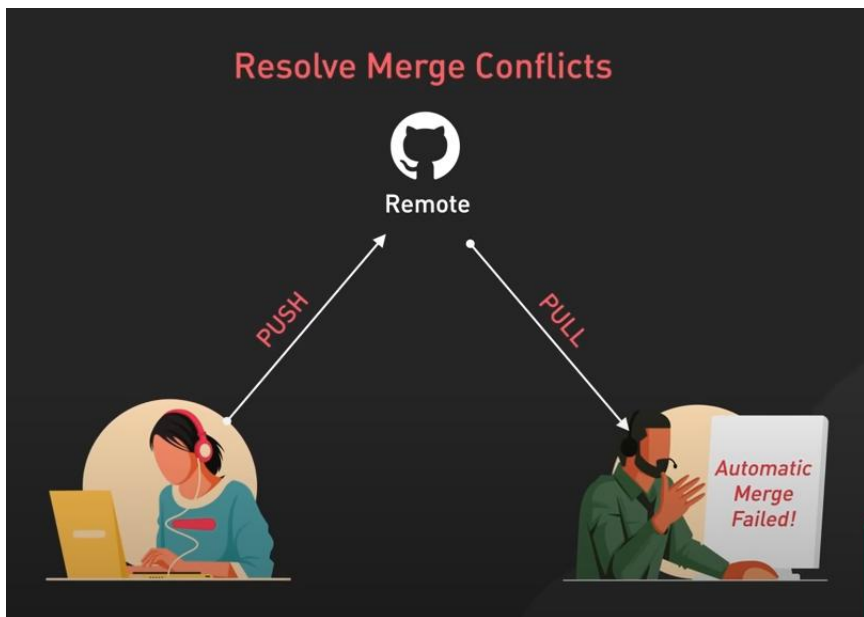
- Komunikacija z oddaljenim repozitorijem poteka dvosmerno
- Podatke lahko tudi povlečemo (git pull) v naš lokalni repozitorij



- Git omogoča, da lahko naše delo poteka v ločenih vejah (branches):
 - Določene funkcionalnosti tako lahko razvijamo ločeno od naše glavne (main) veje
 - Spremembe iz ločenih vej lahko potem združimo nazaj v glavno vejo.



- Med sodelovanjem lahko pride do konfliktov pri spremembah datotek:
 - Take konflikte je potrebno razrešiti preden lahko svoje spremembe pošljemo nazaj v repozitorij.



```
colors.txt x
src > colors.txt
1 red
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2 <<<<<<< HEAD (Current Change)
3 green
4 =====
5 white
6 >>>>>>> his-branch (Incoming Change)
7 blue
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
react-app-demo my-branch git merge his-branch
Auto-merging src/colors.txt
CONFLICT (content): Merge conflict in src/colors.txt
Automatic merge failed; fix conflicts and then commit the result.
x react-app-demo my-branch >M<
```

SOURCE CONTROL

Message (%Enter to commit on...)

Commit

Changes

1 README.md Aggre... + U

Methods_analysis.ipynb

bcdb5f2d - Remove weighted avg ... (4 files) X

README.md U

Aggregation_dev.ipynb

Aggregation/Notebooks

2895 </tr>\n",

2896 " </tbody>\n",

2897 "</table>\n",

2898 "<p>5 rows 25 columns</p>\n",

2899 "</div>"

2900],

2901 "text/plain": [

39 hidden lines

2941 "3 3.0

2942 "4 3.0

2943 "\n",

2944 "

2945 "0 {'Text_ID': 'ParlaMint-

2946 "1 {'Text_ID': 'ParlaMint-

2947 "2 {'Text_ID': 'ParlaMint-

2948 "3 {'Text_ID': 'ParlaMint-

2949 "4 {'Text_ID': 'ParlaMint-

2950 "\n",

2951 " diff_mean diff_median

2952 "0 0.838881 0.636969

2953 "1 0.838881 0.636969

2954 "2 0.838881 0.636969

2955 "3 0.491818 0.340636

2956 "4 0.491818 0.340636

2957 "\n",

2958 " diff_weight length l

2959 "0 0.838881 4

2960 "1 0.838881 11

2961 "2 0.838881 5

2962 "3 0.491818 2

2963 "4 0.491818 4

2964 "\n",

2965 "[5 rows x 25 columns]"

2966]

2967 },

2968 "execution_count": 18,

2969 "metadata": {},

2970 "output_type": "execute_resu

2971 }

9 hidden lines

File Edit View Repository Branch Help

Let's get started!

Add a repository to GitHub Desktop to start collaborating

Create a tutorial repository...

Clone a repository from the Internet...

Create a New Repository on your hard drive...

Add an Existing Repository from your hard drive...

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

Filter your repositories

Your repositories

katjameden/ParlaMint

katjameden/ParlaMint_Sentiment

katjameden/siParl

nikcesenjv/Diploma

clarin-eric

clarin-eric/ParlaMint

clarinsi

clarinsi/audio_validation

clarinsi/babushka-bench

clarinsi/benchich

clarinsi/bilateral-srl

clarinsi/cla

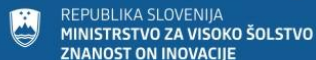
clarinsi/clarin-dspace

clarinsi/CLARIN.SI-crawling-pipeline

Clone katjameden/ParlaMint_Sentiment

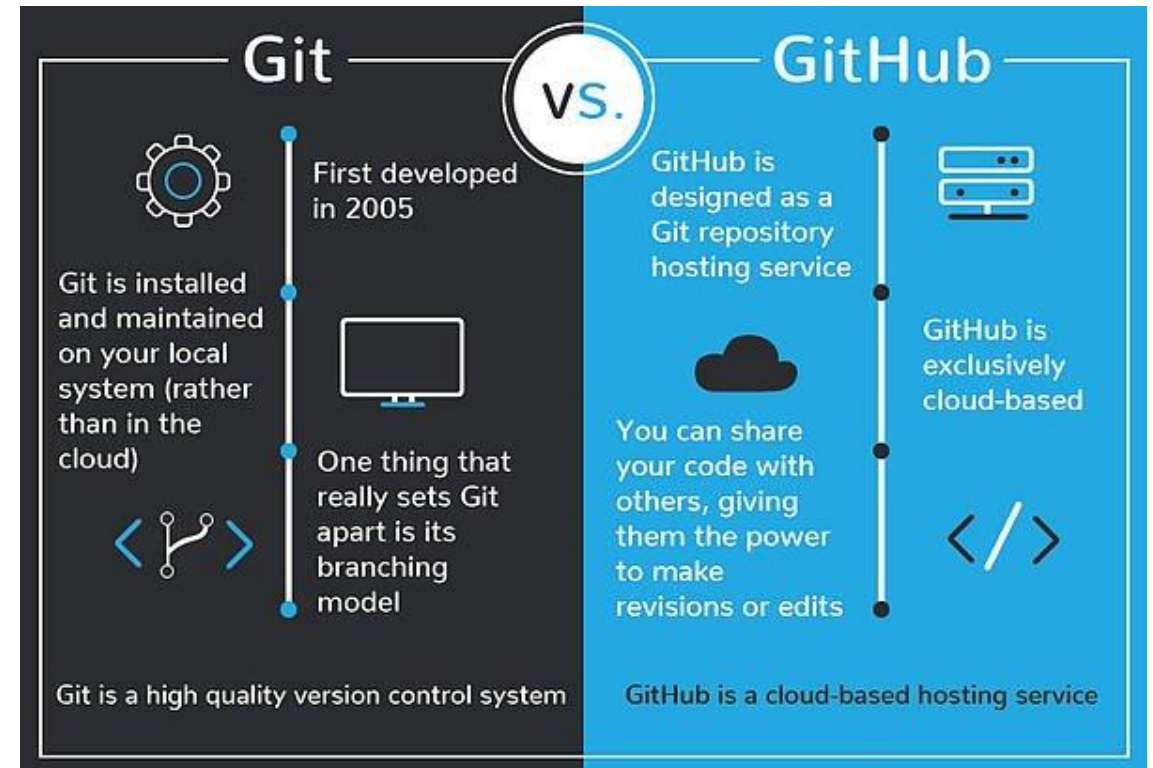


Git, GitHub in GitLab



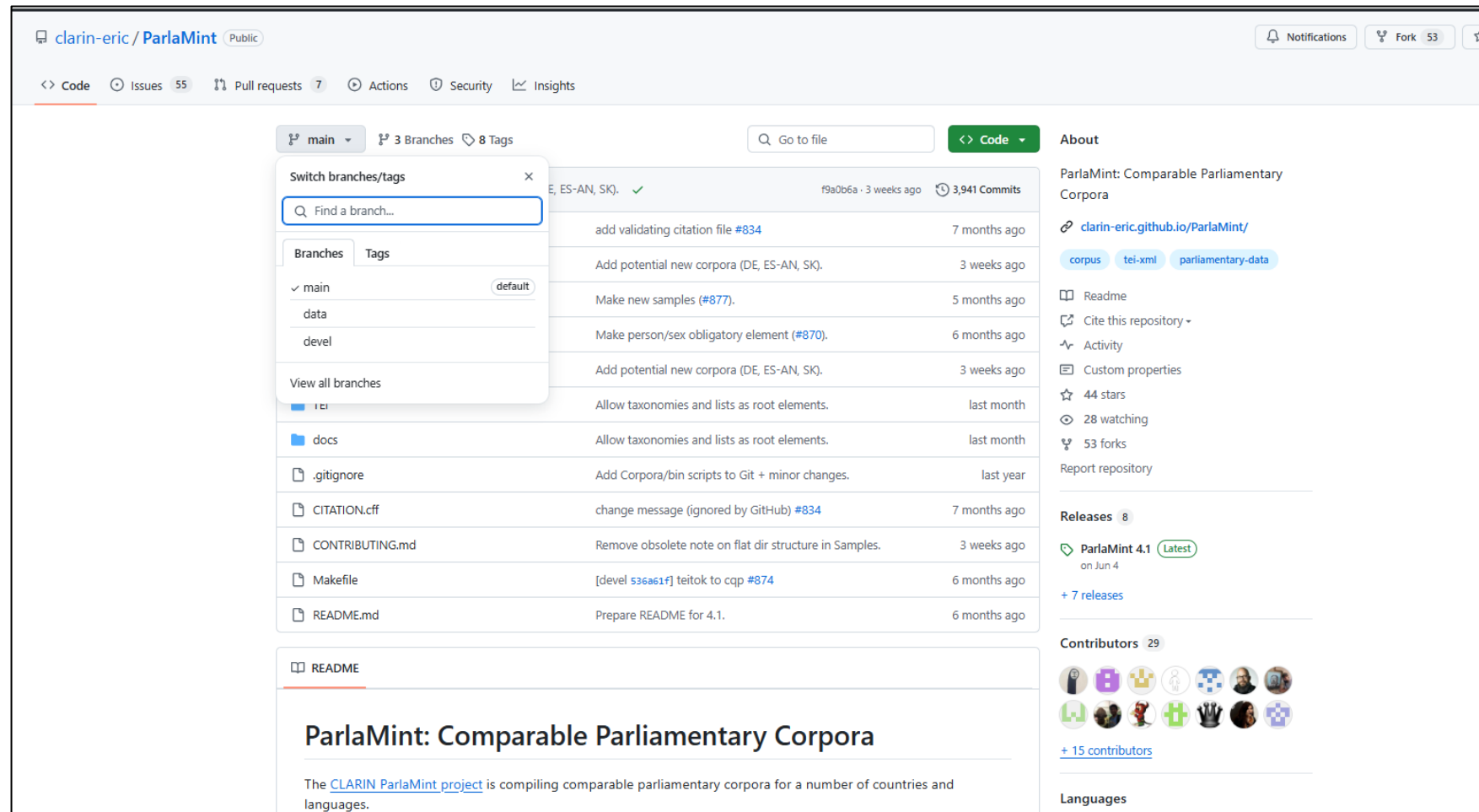
Git vs. GitHub – ..."GitHub is to Git, what Facebook is to your actual face"

- Spletna storitev, ki omogoča gostovanje Git repozitorijev
- Omogoča:
 - Sodelovanje med razvijalci
 - Hrambo in delitev našega dela z drugimi
 - Uporabniški (grafični) vmesnik, ki olajša delo z repozitoriji
 - Lažje pregledovanje kode (posameznih commit-ov, vej, mejnikov in sprememb v kodi) v primerjavi z Gitom
 - Dodatne funkcionalnosti, ki nadgradijo osnovne funkcije Gita



GitHub funkcionalnosti

- [GitHub Issues](#) (pogovori o in sledenje napakam, možnih izboljšavah itd. v repozitoriju/kodi, navezava na posamezne committe)
- [GitHub Pages](#) (možnost dodajanja dokumentacije)
- [CI/CD, GitHub Actions](#): omogočajo avtomatizacijo in izvedbo delotokov (workflows) za razvoj programske opreme



Commit ce1857f

katjameden committed on Jan 16, 2023

[Preview](#) [Give feedback](#) [Browse files](#)

Implement SI feedback corrections.

main (#485) · v4.1 · v3.0-en

1 parent [7620efb](#) commit ce1857f

Filter files...

- Data/ParlaMint-SI
 - ParlaMint-SI-listOrg.xml
 - ParlaMint-SI.ana.xml
 - ParlaMint-SI.xml
 - ParlaMint-SI_2014-08-25-S...
 - ParlaMint-SI_2014-08-25-S...
 - ParlaMint-SI_2018-06-22-S...
 - ParlaMint-SI_2018-06-22-S...
 - ParlaMint-SI_2022-04-06-S...
 - ParlaMint-SI_2022-04-06-S...

9 files changed +193 -80 lines changed

Data/ParlaMint-SI/ParlaMint-SI-listOrg.xml				+10 -9	...
↑	@@	-366,19	+366,18	@@	
366	366		active="#party.SAB"		
367	367		passive="#party.ZaSLD"		
368	368		when="2017-10-07"/>		
369	-		<relation name="renaming" mutual="#party.SMC.2 #party.GAS" when="2021-12-04"/>		
370	-		<relation name="coalition"		
371	-		mutual="#party.PS #party.SD #party.DL #party.DeSUS"		
372	-		from="2013-03-20"		
373	-		to="2014-09-18"		
374	-		ana="#GOV.11"/>		
369	+		<relation name="renaming"		
370	+		active="#party.Konkretno"		
371	+		passive="#party.SMC.2 #party.GAS"		
372	+		when="2021-12-04"/>		
375	373		<relation name="coalition"		
376	374		mutual="#party.SMC.1 #party.SMC.2 #party.SD #party.DeSUS"		
377	375		from="2014-09-18"		
378	376		to="2018-09-12"		
379	377		ana="#GOV.12"/>		
380	378		<relation name="opposition"		
381	-		mutual="#party.SDS.2 #party.IMNS #party.ZaAB #party.NeP #party.NP #party.NSi #party.Levica.1 #party.Levica.2"		
379	+		active="#party.SDS.2 #party.IMNS #party.ZaAB #party.NeP #party.NP #party.NSi #party.Levica.1 #party.Levica.2"		
380	+		passive="#GOV"		

GitHub Issues

Filters

is:issue is:open

Labels 13

Milestones 1

New issue

55 Open 431 Closed

Author Label Projects Milestones Assignee Sort

Per-country translations in taxonomies bug Taxonomy

#880 opened 3 weeks ago by TomazErjavec Future 6

GB: ministers missing bug

#879 opened on Oct 5 by TomazErjavec Future

IT: notes inside paragraphs are splitting paragraphs bug

#872 opened on May 14 by matyaskopp Future

BA: incorrect forename/surname distinction bug

#852 opened on Mar 5 by TomazErjavec Future

LV: missing first note in TEI.ana bug

#848 opened on Feb 29 by matyaskopp 2 tasks Future

LV: all seg are merged into first one in TEI.ana bug

#847 opened on Feb 29 by matyaskopp Future

validate-parlamint speedup enhancement

#846 opened on Feb 28 by matyaskopp Future

Per-country translations in taxonomies #880

Open TomazErjavec opened this issue last month · 6 comments

TomazErjavec commented last month

Collaborator

Currently we have and support multilingual taxonomies but it can happen (esp. in the [legislature taxonomy](#)) that the translation of a term for one country differs from the translation for another, even though both use the same language (e.g. AT and DE).

For this reason the language of a translation should be - when needed - extended by the country code, e.g. de-AT vs. de-DE. For this, all the code dealing with `@xml:lang` (at least in connection with taxonomies, but ideally everywhere) would have to be revised, as well as the taxonomies themselves.

This issue becomes even more relevant if we were to add country-specific hyperlinks to Wikipedia as part of the description for particular categories.

TomazErjavec added bug Taxonomy labels last month

TomazErjavec added this to the **Future** milestone last month

TomazErjavec assigned **matyaskopp** and **TomazErjavec** last month

matyaskopp commented last month

Collaborator

I checked the language tag documentation: <https://www.rfc-editor.org/rfc/rfc5646.html#page-5>, and it contains a more detailed structure than I expected.
I can see a problem with using the `-region` part of the language tag because it allows only 2-letter values (ISO 3166-1 code) or some 3-digit encoding that I don't know (UN M.49 code), so the value `es-ES-CT` is invalid.

It can probably be hacked with extensions `es-ES-a-CT` (<https://www.rfc-editor.org/rfc/rfc5646.html#section-2.2.6>) or private subtags `es-ES-x-CT` (<https://www.rfc-editor.org/rfc/rfc5646.html#section-2.2.7>). I hope this situation will not happen because it will complicate things even more...

Assignees

matyaskopp

TomazErjavec

Labels

bug Taxonomy

Projects

None yet

Milestone

Future

Development

Create a branch for this issue or link a pull request.

Notifications

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

Pin issue

Transfer issue

Erjavec, T., et al. (2024). *ParlaMint: Comparable Parliamentary Corpora*. Dostopno na: <https://github.com/clarin-eric/ParlaMint>

GitHub Pages

The structure and encoding of [ParlaMint corpora](#)

2024-05-14

1. Introduction

This document is meant to serve as a reference for the encoding of ParlaMint corpora of parliamentary proceedings. In order for the ParlaMint corpora to be interoperable (i.e. so that the same scripts can be used to process them), their structure is fairly rigid, both in terms of file names and folder structure, as well as their TEI XML encoding. This is not to say that all the corpora have to contain exactly the same information because we distinguish obligatory information, which all the corpora should contain, from that which is optional, and present only in the corpora for which it has been possible to gather it from the corpus sources.

This document is a specialisation of [Parla-CLARIN](#), itself a customisation of the [TEI Guidelines](#). But while Parla-CLARIN gives fairly general recommendations for encoding corpora of parliamentary proceedings, ParlaMint, as mentioned, is much stricter. This document gives very specific encoding recommendations without necessarily stating the reasons for their choice. It covers the overall structure of ParlaMint corpora, the metadata they contain, the encoding of transcriptions, and, for the linguistically annotated version, the encoding of word-level linguistic annotations, syntactic dependencies and named entities.

The document is not meant as a tutorial on TEI or ParlaMint, but as a reference to elements, their nesting and attributes exemplified by snippets from the existing ParlaMint corpora. Other sources can help in understanding the encoding of ParlaMint corpora:

- The freely available paper:
Erjavec, T., Ogrodniczuk, M., Osenova, P. et al. The ParlaMint corpora of parliamentary proceedings. Language Resources & Evaluation (2022). <https://doi.org/10.1007/s10579-021-09574-0>.
- The [Parla-CLARIN](#) guidelines, which provide general guidelines for encoding parliamentary corpora in TEI; they also give links to the relevant chapters of the [TEI Guidelines](#).
- Samples of ParlaMint corpora, available in the Data/ directory of the ParlaMint GitHub repository, esp. useful as they give the complete picture of a ParlaMint corpus; note that the samples in the [main branch](#) are supposed to be publication-ready, while those in the [data branch](#) are work in progress.

The rest of these recommendations are structured as follows:

- [Chapter 2](#) explains the overall XML structure of a ParlaMint corpus, and introduces the distinction between the corpus root and corpus components;
- [Chapter 3](#) explains some general requirements and the file-naming conventions a ParlaMint corpus has to meet; it also introduces the top level elements and their attributes and the main pointing attributes;
- [Chapter 4](#) concentrates on the structure and encoding of the corpus metadata, such as the title information, documenting the source of the corpus, taxonomies used etc.;
- [Chapter 5](#) explains how and what information must be encoded about the persons giving the speeches and the (political) organisations they belong to;
- [Chapter 6](#) treats the encoding of the transcripts, including speeches and transcriber notes;
- [Chapter 7](#) details the addition of linguistic annotations to the corpus;
- [Chapter 8](#) introduces scripts to finalise, validate and convert a ParlaMint corpus to other formats;
- [Chapter 9](#) gives instructions on how to contribute samples of a ParlaMint corpus to GitHub;
- [Appendix A](#) gives the formal specification of the Parla-CLARIN schema.

2. Overall corpus structure

2.1. XML structure

The parliamentary proceeding of one country or autonomous region constitute one ParlaMint corpus, which is stored as one XML document, with [<teiCorpus>](#) as its top-level element. It is composed of a [<teiHeader>](#), giving the metadata for the corpus as a whole (further detailed in the Section on [Corpus metadata](#)), followed by a series of [<TEI>](#) elements that each contain one *corpus component*, as illustrated¹ below:

```
<!-- Corpus root -->
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <TEI>...</TEI> <!-- Corpus component -->
  <TEI>...</TEI> <!-- Corpus component -->
  ...
  <!-- More corpus components -->
</teiCorpus>
```

GitHub in GitLab

- GitLab je odprtokodna platforma za celovit razvoj programske opreme podoben servis kot GitHub.
- Omogoča:
 - Gostovanje repozitorijev Git (in nadzor različic), ki pa je le del platforme
 - Omogoča sledenju težavam, pregledovanje kode, analize repozitorija, delotoka...
 - Omogoča boljši nadzor nad dostopom do repozitorija in boljšo varnost
- V nasprotju z GitHub-om:
 - GitLab je odprtokoden
 - Inštaliramo si lahko svoj GitLab, z GitHub-om to ni možno



Difference between GitLab and GitHub:

Parameters	GitLab	GitHub
Developed by	GitLab was developed by Dmitriy Zaporozhets and Valery Sizov.	GitHub was developed by Chris Wanstrath, Tom Preston-Werner, P. J. Hyett, and Scott Chacon.
Open-sourced	GitLab is open-source for community edition.	GitHub is not open source.
Public Repository	It allows users to make public repository.	It allows users to have unlimited free repository.
Private Repository	GitLab provides free private repository.	GitHub also allows users to have free private repository with unlimited collaborators.
Navigation	GitLab provides the feature of navigation into the repository.	GitHub allows users to navigate usability.
Project Analysis	GitLab provides user to see project development charts.	GitHub doesn't have this feature yet but they can check the commit history.
Advantages	• GitLab is freely available and is open source for community edition • It is a cloud-native application and is highly secure.	• It helps us create an organized document for the project. • It is used for sharing the work in front of the public.
Disadvantages	• GitLab is available with many bugs and it makes the user experience sloppy. • It is difficult to manage code reviews for first-timers.	• It is not open-source. • It supports only Git version control.
Company	It is owned by GitLab Inc.	It is owned by Microsoft Corporation.
Security	More secure than Github.	It is less secure as security Dashboard, License Compliance is missing in GitHub.
Attachments	Gitlab supports adding other types of attachments.	GitHub does not allow adding other types of attachments.

Search or go to...

Project

clarinsi-cqp

Pinned

Issues2

Merge requests0

Manage

Plan

Code

Build

Deploy

Operate

Monitor

Analyze

clarinsi / clarinsi-cqp

Management of corpora on CLARIN.SI concordancers

Update Tredndi.
Tomaž Erjavec authored 2 days ago

d9381c43

master

clarinsi-cqp

+

History

Find file

Edit

Code

README

Name	Last commit	Last update
backups	Redmine #2393: switched to repo-internal backups dire...	1 year ago
config	Correct wrong config.	1 month ago
generated	Redmine #2393: continued with shift from regenerating ...	2 years ago
registry	Update Tredndi.	2 days ago
scripts	Gitlab #10: added script for checking compatibility of re...	7 months ago
subcorp	Make subcorp folder with test entry (#11)	1 month ago
.gitignore	Redmine #2393: made backup permission change more...	1 year ago
00BUGS.txt	Change ParlaMint 3.1 (and 3.0) to 4.0.	1 year ago

Search or go to...

Project

clarinsi-cqp

Pinned

Issues2

Merge requests0

Manage

Plan

Code

Build

Deploy

Operate

Monitor

clarinsi / clarinsi-cqp / Repository Analytics

Repository Analytics

Programming languages used in this repository

Measured in bytes of code, Excludes generated and vendored code.

Percentage

Python

Makefile

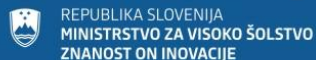
Shell

XSLT

Perl

Used programming language

Povzetki in zaključki



Git, GitHub in GitLab

- Git je najbolj popularen distribuiran sistem nadzora različic in omogoča sledenje spremembam v kodi/datotekah
 - Lokalni sistem, sodelovanje med razvijalci
 - Fleksibilen sistem, vendar tudi precej kompleksen
- Razumevanje delotoka sistema Git je ključen za uspešno delo:
 - Git je možno uporabljati na različne načine
- GitHub je spletni servis, ki še olajša sodelovanje med razvijalci:
 - Uporabniku prijazen vmesnik
 - Lažji pregled nad kodo in posameznimi commiti
 - Omogoča dodatne funkcionalnosti kot so GitHub Issues, Pages in Actions
- GitLab je podoben GitHubu, vendar se v nasprotju z GitHubom osredotoča na celovit razvoj programske opreme:
 - Nadzor različic in gostovanje repozitorijev Git sta le del platforme
 - Omogoča dodatne funkcije (analize repozitorija, delotoka...)
 - Omogoča boljši nadzor nad dostopom in boljšo varnost

Priročne povezave in reference:

Reference/Literatura:

- Chacon, S. and Straub, B. (2024). *Pro Git – Everything you need to know about Git*. Dostopno na: <https://git-scm.com/book/en/v2>
- Difference Between GitLab and GitHub. (n.d.). Dostopno na: <https://www.geeksforgeeks.org/difference-between-gitlab-and-github/>
- Git Reference (n.d.). Dostopno na: <https://git-scm.com/docs>
- GitHub. (n.d.). GitHub [Docs](https://docs.github.com/en). Dostopno na: <https://docs.github.com/en>
- Kaushik, G. (2020). The Ultimate Git & GitHub Cheatsheet. Dostopno na: <https://geetikakaushik2020.medium.com/the-ultimate-git-github-cheatsheet-fe7f30f8ede>
- Kelmendi, D. (2017). *Git — the superstar version control system*. Dostopno na: <https://medium.com/remote-engineering-academy/git-the-superstar-version-control-system-2382951dfc15>
- The Best Git Books. (n.d.). Dostopno na: <https://github.com/maniramakumar/the-best-git-books/tree/master>

Video posnetki /Tutorials:

- Hamedani, M. (2020). *Git Tutorial for Beginners: Learn Git in 1 Hour*. Dostopno na: <https://www.youtube.com/watch?v=8JJ101D3knE>
- Lam, S. (2023). *How Git Works: Explained in 4 Minutes*. Dostopno na: https://www.youtube.com/watch?v=e9lmsKot_SQ
- W3C Schools. (n.d.). *Git Tutorial*. Dostopno na: <https://www.w3schools.com/git/>

Kodni urejevalniki in Git Integracije:

- Le Moment It. (2023). *What is the best GIT client?* Dostopno na: https://medium.com/@le_moment_it/what-is-the-best-git-client-3e394c6c1b4d
- Using Git source control in VS [Code](https://code.visualstudio.com/docs/sourcecontrol/overview). (n.d.). Dostopno na: <https://code.visualstudio.com/docs/sourcecontrol/overview>

Primeri v predstavitvi:

- Erjavec, T., et al. (2024). *ParlaMint: Comparable Parliamentary Corpora*. Dostopno na: <https://github.com/clarin-eric/ParlaMint>
- Erjavec, T., et al. (2024). *The structure and encoding of ParlaMint corpora*. Dostopno na: <https://clarin-eric.github.io/ParlaMint>

Centralna tehniška knjižnica Univerze v Ljubljani
Central Technical Library at the University of Ljubljana
Trg republike 3, SI-1000 Ljubljana
Slovenija / Slovenia

Uredniški odbor:

dr. Tea Romih, CTK (odgovorna urednica)
Barbara Bradaš Premrl, ZRS Koper
Brina Klemenčič, UKM
dr. Mojca Kotar, UL
dr. Miha Peče, ZRC SAZU

Hvala za pozornost!

