Introducing Effective Information Environment Organizational Strategies (EIEIOS) through GitLab for Non-Programmers

90th MORS Symposium

An Analyst's Perspective on Reframing Knowledge Management

13-16 June 2022

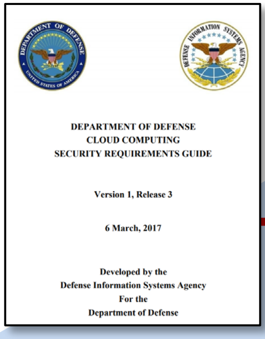
DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited. Release of this information in this brief does not imply any commitment or intent on the part of the U.S. Government to provide any additional public information on any topic presented herein or to participate in any wargame or experiment.

Purpose & Introduction

Purpose: Share vision (and key application/example) of an emerging asynchronous sustainable distributed community tool development effort.

Agenda

- KM: Challenges and Opportunities.
- Terminology (good, bad, and ugly).
- Vision of Distributed Work.
- GitLab Approach:
  - *Platform Search.*
  - *Value Added Examples.*
  - *Institutional / Organizational Advantages.*
- Individual, Institutional, Continuity Challenges.



DEPARTMENT OF DEFENSE
CLOUD COMPUTING
SECURITY REQUIREMENTS GUIDE

Version 1, Release 3

6 March, 2017

Developed by the
Defense Information Systems Agency
For the
Department of Defense

Army Futures Command * Modernization Application and Data Environment
Modernization
Application
Data

A Few Caveats...

- Software programming vs statistical programming.
  - *Operations Researchers will (at some point) produce stand-alone code/products.*
- Oriented towards analysts who have an interest in:
  - *Seeing "what's out there."*
  - *Joining the community.*
  - *Improving existing methods.*
  - *Technical collaboration as a complement to interpersonal coordination.*

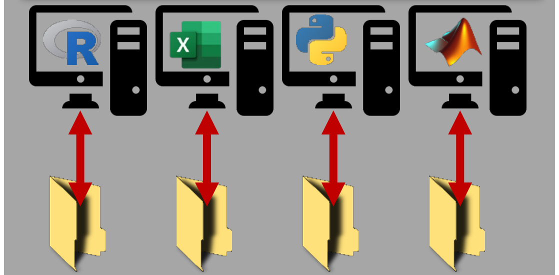
Focused on GitLab as a "*small win*" – not a panacea.

*How does a vision of **continuous, decentralized tool development** among non-programmers support distributed work through GitLab?*

Background Experience

How do we (non-programmers) develop and sustain tools?

Current Approach



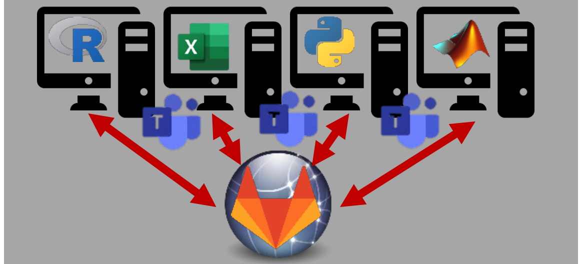
What we're used to:

- Single person, single machine coding. *"It's saved on my local machine."*
- Point to point, single-time deliberate & good-faith transfers. *"Let me email it to you, and I'll walk you through it later on."*
- Person-based recall. *"What do we have? Let me ask SME."*

Thought Experiments

- **Leader Awareness:** A new LTC is trying to get a sense of "what tools have we built" here.
- **Talent Resonance:** A recent CIV hire does not know MDO, but they do know Java.
- **Sustainability:** A team builds a great tool, but the project is over. What (ideally) happens?
- **Skill Building:** A novice analyst is toiling away to create a function that's already been implemented.
- **Prof. Development:** Will a new CPT have a competitive advantage in their next assignment if they can build quick tools.
- **Institutional Strength:** An experienced analyst is interested in improving, advising, and technically advancing other grass-roots efforts.

Emerging Approach



Where we're heading:

- Multi-machine, video-shared team coding. *"Can you check me as I do this live?"*
- Real-time sync. *"Can I pull it? Give me 30 s."*
- Undirected download. *"I saw the code you posted, can you walk me through lines 25-30?"*
- System-based recall. *"What kinds of projects do a specific thing?"*

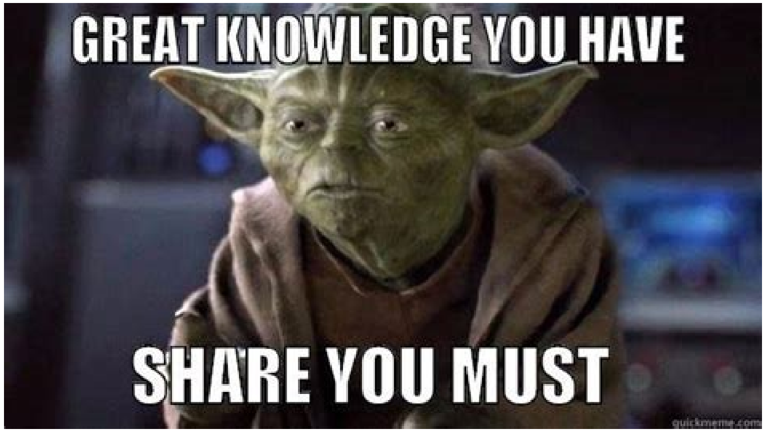
How do we pursue a 21st century approach to collaborative tool development?

Sharing Information

A Known Challenge

We know we wanted something to help the community build, share, and sustain tools. Is this "Knowledge Management?"

"Our founders built this company on a certain set of principles that continue to shape everything we do. I really hope someone wrote them down before they retired."



"NO YOU CAN'T ASK A QUESTION."

What is Knowledge Management?

The Best Definitions Lack Helpful Clarity

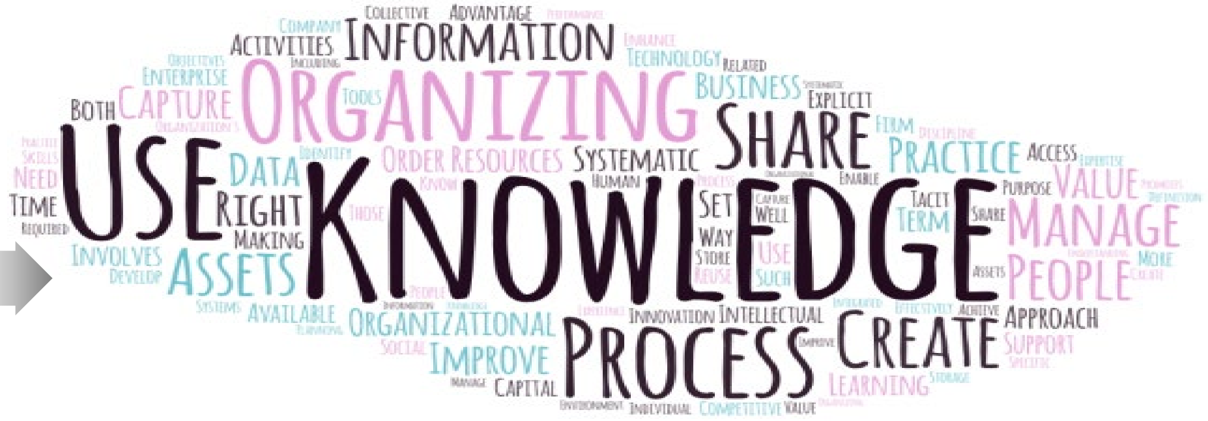
Girard, J.P., & Girard, J.L. (2015). Defining knowledge management: Toward an applied compendium, *Online Journal of Applied Knowledge Management*. 3(1), 1-20

*Took 100+ KM definitions "…some are from academics, while others are from practitioners**, some are from the government,** others from the for-profit sector, and still others are from the not for profits…."*

Conducted a frequency analysis and determined that the most common words across definitions included:

***knowledge**, **organization**, **process**, **information**, **use**, **share**, **create**, and **manage**.*

Retrieved from: https://www.johngirard.net/km/



Two Winners Emerge:

–**US Army ATP 6-01.1:** "*Knowledge management (KM) is the process of enabling knowledge flow to enhance shared understanding, learning, and decision making.*"

–**Meta-analysis (Girard, 2015):** *"Knowledge Management is the process of creating, sharing, using, and managing the knowledge and information of an organization."*

What is Knowledge Management?

The Best Definitions Lack Helpful Clarity

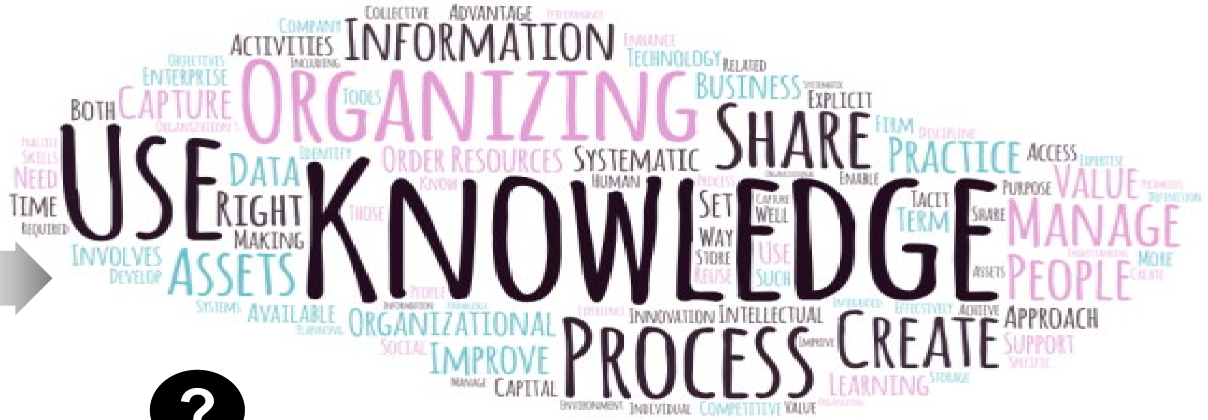
Girard, J.P., & Girard, J.L. (2015). Defining knowledge management: Toward an applied compendium, *Online Journal of Applied Knowledge Management*. 3(1), 1-20

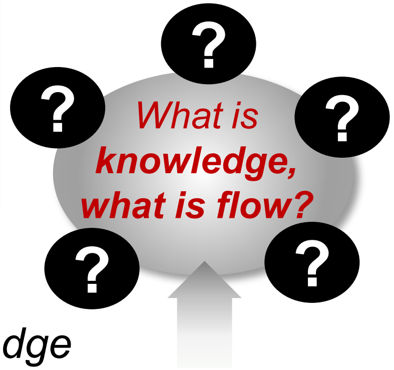
*Took 100+ KM definitions "…some are from academics, while others are from practitioners, **some are from the government,** others from the for-profit sector, and still others are from the not for profits…."*

Conducted a frequency analysis and determined that the most common words across definitions included:

***knowledge**, **organization**, **process**, **information**, **use**, **share**, **create**, and **manage**.*

Retrieved from: https://www.johngirard.net/km/





What is **knowledge, what is flow?**

Two Winners Emerge:

–**US Army ATP 6-01.1:** "*Knowledge management (KM) is the process of enabling knowledge flow to enhance shared understanding, learning, and decision making.*"

–**Meta-analysis (Girard, 2015):** *"Knowledge Management is the process of creating, sharing, using, and managing the knowledge and information of an organization."*

What is Knowledge Management?

The Best Definitions Lack Helpful Clarity

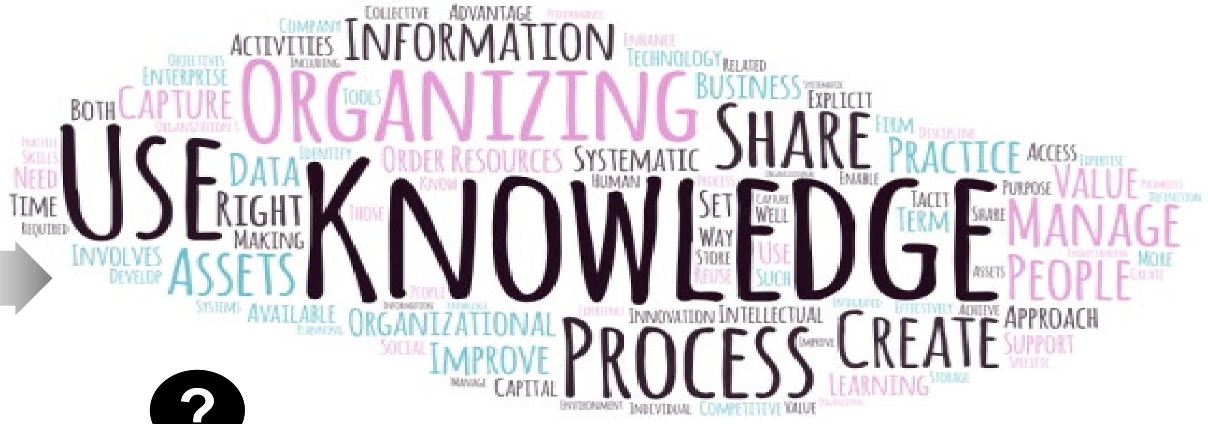
Girard, J.P., & Girard, J.L. (2015). Defining knowledge management: Toward an applied compendium, *Online Journal of Applied Knowledge Management*. 3(1), 1-20

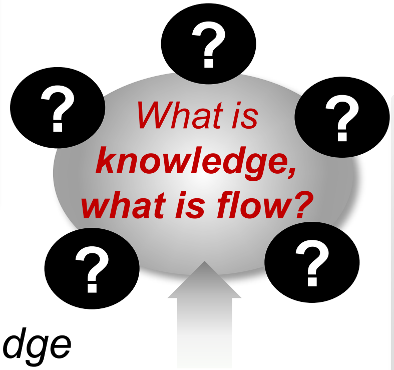
*Took 100+ KM definitions "…some are from academics, while others are from practitioners, **some are from the government,** others from the for-profit sector, and still others are from the not for profits…."*

Conducted a frequency analysis and determined that the most common words across definitions included:

***knowledge**, **organization**, **process**, **information**, **use**, **share**, **create**, and **manage**.*

Retrieved from: https://www.johngirard.net/km/

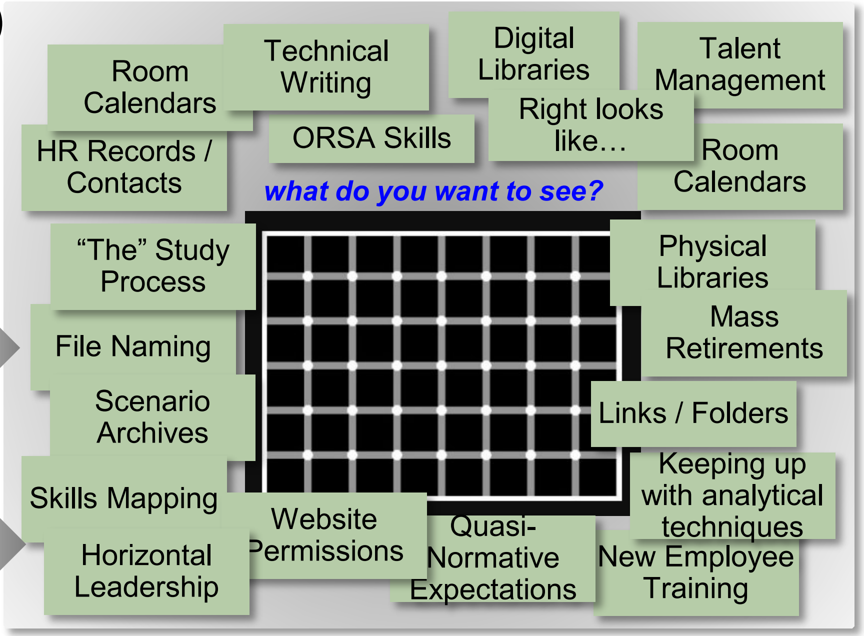




Two Winners Emerge:

– **US Army ATP 6-01.1:** "*Knowledge management (KM) is the process of enabling knowledge flow to enhance shared understanding, learning, and decision making.*"

– **Meta-analysis (Girard, 2015):** "*Knowledge Management is the process of creating, sharing, using, and managing the knowl and information of an organization.*"


Room Calendars
Technical Writing
Digital Libraries
Talent Management
HR Records / Contacts
ORSA Skills
Right looks like…
Room Calendars
what do you want to see?
"The" Study Process
Physical Libraries
File Naming
Mass Retirements
Scenario Archives
Links / Folders
Skills Mapping
Keeping up with analytical techniques
Website Permissions
Quasi-Normative Expectations
Horizontal Leadership
New Employee Training

What is Knowledge Management?

The Best Definitions Lack Helpful Clarity

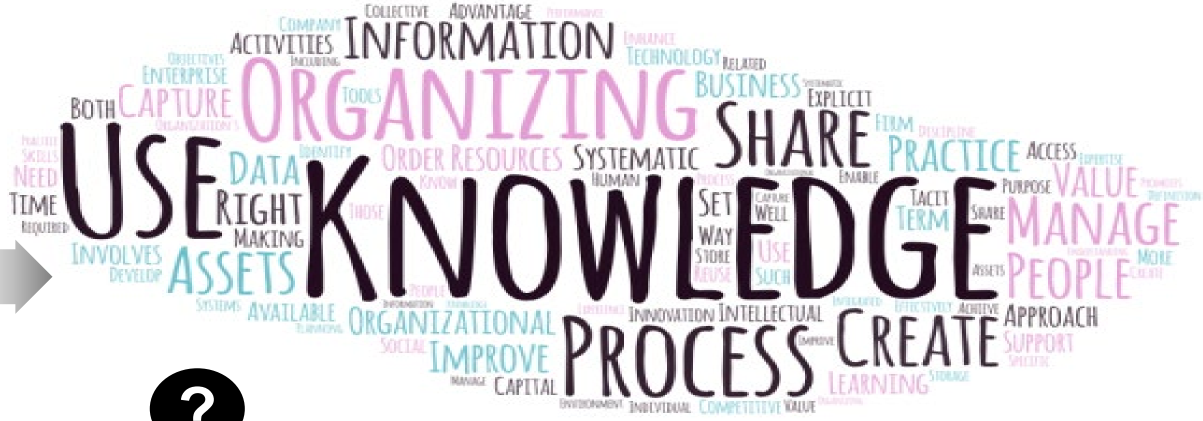
Girard, J.P., & Girard, J.L. (2015). Defining knowledge management: Toward an applied compendium, *Online Journal of Applied Knowledge Management*. 3(1), 1-20

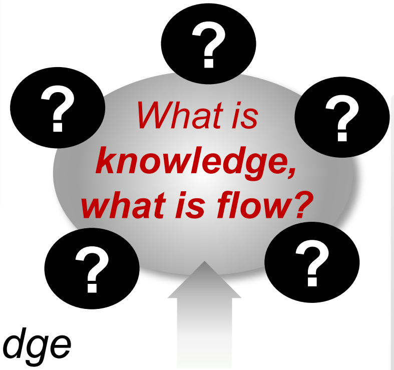
*Took 100+ KM definitions “…some are from academics, while others are from practitioners, **some are from the government,** others from the for-profit sector, and still others are from the not for profits….”*

Conducted a frequency analysis and determined that the most common words across definitions included:

***knowledge**, **organization**, **process**, **information**, **use**, **share**, **create**, and **manage**.*

Retrieved from: https://www.johngirard.net/km/





*What is **knowledge, what is flow?***

Two Winners Emerge:

–**US Army ATP 6-01.1:** “*Knowledge management (KM) is the process of enabling knowledge flow to enhance shared understanding, learning, and decision making.*”

–**Meta-analysis (Girard, 2015):** *“Knowledge Management is the process of creating, sharing, using, and managing the knowl and information of an organization.”*

“You keep saying knowledge management. I don’t think it means what you think it means.”

Vision of Collaborative Tool Development

Pursuing Alternate Guiding Principles

1 Analytic Integrity

*"there is a specific, extra type of **integrity** that is not lying, but **bending over backwards to show how you're maybe wrong**, that you ought to have when acting as a scientist. And this is our responsibility as scientists, certainly to other scientists, and I think to [the broader public]."*

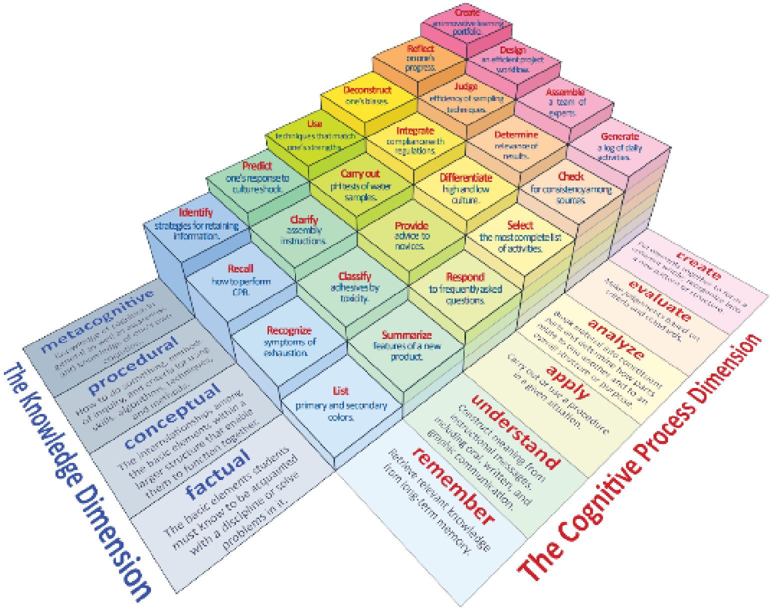
https://sites.cs.ucsb.edu/~ravenben/cargocult.html

ARE YOU SURE THE DATA YOU GAVE ME IS CORRECT?
I'VE BEEN GIVING YOU INCORRECT DATA FOR YEARS. THIS IS THE FIRST TIME YOU'VE ASKED.
Dilbert.com DilbertCartoonist@gmail.com

2 Knowledge & Cognitive Processes

Psychological research of learning, has produced helpful terminology for learning organizations.

…can be…	…in that you know…	…and sounds like…
Factual	Basic elements to solve problems (declarative ***terms***, ***details***, and specifics)	*I know that …*
Conceptual	Interrelationships that enable functions (classes, ***categories***, principles, generalizations, theories, ***models***, structures)	*I know about …*
Procedural	How to do something (i.e. methods, algorithms, ***techniques***, ***processes***, ***steps***, and pot. errors)	*I know how to …*
Meta-Cognitive	Your own status and relation to strategic purpose, vision/intent, and conditional constraints.	*I know why …*


Create an innovative learning portfolio.
Reflect on one's progress.
Design an efficient project workflow.
Deconstruct one's biases.
Judge efficiency of sampling techniques.
Assemble a team of experts.
Use techniques that match one's strengths.
Integrate compliance with regulations.
Determine relevance of results.
Generate a log of daily activities.
Predict one's response to culture shock.
Carry out pH tests of water samples.
Differentiate high and low culture.
Check for consistency among sources.
Identify strategies for retaining information.
Clarify assembly instructions.
Provide advice to novices.
Select the most complete list of activities.
Recall how to perform CPR.
Classify adhesives by toxicity.
Respond to frequently asked questions.
Recognize symptoms of exhaustion.
Summarize features of a new product.
List primary and secondary colors.
metacognitive Knowledge of cognition in general as well as awareness and knowledge of one's own cognition.
procedural How to do something, methods of inquiry, and criteria for using skills, algorithms, techniques, and methods.
conceptual The interrelationships among the basic elements within a larger structure that enable them to function together.
factual The basic elements students must know to be acquainted with a discipline or solve problems in it.
The Knowledge Dimension
create Put elements together to form a coherent whole; reorganize into a new pattern or structure.
evaluate Make judgements based on criteria and standards.
analyze Break material into constituent parts and determine how parts relate to one another and to an overall structure or purpose.
apply Carry out or use a procedure in a given situation.
understand Construct meaning from instructional messages, including oral, written, and graphic communication.
remember Retrieve relevant knowledge from long-term memory.
The Cognitive Process Dimension

A Model of Learning Objectives–based on *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives* by Rex Heer, Center for Excellence in Learning and Teaching, Iowa State University is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

IOWA STATE UNIVERSITY
Center for Excellence in Learning and Teaching

Introducing EIEIOS

New Terminology for when Information Matters

- **E**ffective **I**nformation **E**nvironment **O**rganizational **S**trategies (**EIEIOS;** EIOS, for short) should **improve performance.**
- ***"Information Environment,"*** is more like the way we naturally, think, talk, and do:
  - You can navigate the environments more or less easily, given better or worse maps, experience, communication, inter-personal trust, and ongoing efforts to "reshape" and remove obstacles in that environment.

This distinction produces a meaningful difference in the clarity of discourse. "KM" creates unnecessary confusion that "EIEIOS" naturally avoids.

Prompt	KM Says…	EIEOS Says…
What is knowledge?	Knowledge can be **tacit** (unstated, a product of experience) or **explicit** (e.g. books, files, dictionaries, publications). *Who knows the dictionary?*	Knowledge is **demonstrable in context**. It underpins cognitive processes that are: factual, conceptual, procedural, or meta-cognitive. *People know things that they use to do something.*
Who does it?	It's a **management process** → knowledge managers enable knowledge flow. *Where do I flow or transfer my knowledge?*	It's a set of **organizational strategies** → every person should work to improve (aka "reshape") the information environment. *What can I do to improve the information around me?*
What are some guiding questions?	- Is the knowledge tacit or explicit? - What issues does this knowledge address? - Where does this knowledge come from? - What does the individual need to know? *Who has the authority to implement changes?*	- What kind of knowledge do we want? - How do we demonstrate our knowledge? - What is the most expedient way us for to learn…? - What information would help me do my job? *What obstacles around me will I remove?*

EIEIOS is (1) *light-weight*, (2) *focused on promoting decentralized participation,* (3) *more likely to prompt small "wins"* (each having distinct constraints / or skills), (4) *less verbally confusing,* and (5) *more like the way we actually think & talk.*

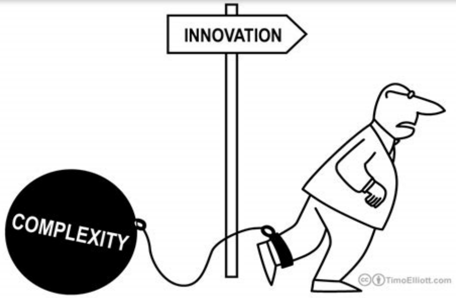
Applying EIEIOS

Pursuing an Impactful Community of Open Development

*Community should be both **culturally resilient** and **technically sustainable.***

Culture

"Culture change. Now that's hard."



- **Open**
  - Ready *access* to technical infrastructure (i.e. minimal steps from in the door to on the site)
  - *Everyone can participate* (all build community)
  - Default *awareness* guaranteed (permissions promote seeing what exists).
- **Development**
  - Innovation – "*change*." Manifested through fruitful partnerships of institutional power and creative energy.
  - Organization – *structure is self-evident,* with "breadcrumbs" provided through clear requirements.
- **Impact**
  - Rapid on-boarding – remove duplicative struggles.
  - Continuity – no artificial sequestration.
  - *Self-directed Talent Management* – limiting factor is the individual.

Technical

"The DOD missed cloud. Let's just let that sink in for a moment…"

- **Flexible**: Version control across all file types (accessible via GUI and Command line).
- **Open**: Accessible without a pre-determined "use-case."
- **Free**: Cannot be license constrained to the point of budgetary or administrative overhead (users, capacity).
- **Available**: Not constrained on unclassified network.
- **Reliable**: Relatively high confidence in platform longevity and technical support (year over year growth).
- **Secure**: Should host at least CUI information (IL4+).
- **Politically tenable**: Must promote positive reputation; cannot "burn bridges" or exhaust social, relational, or organizational capital.
- **Organized**: Resources & structure are self-evident to the point they promote reinforcement by the community.

GitLab presents a viable option to enable the community.

GitLab Presents an Opportunity

GitLab is a "DevOps" solution that enables individuals and groups to actively see, change, synchronize updates, and understand the evolution of (any) shared files.

		FLEXIBLE	OPEN	FREE	AVAIL.	RELIABLE	SECURE	TENABLE	FEASIBLE
1	Naval Postgraduate School *https://gitlab.nps.edu/*	✓	🚫						
2	Modernization Application and Data Environment (MADE) *https://code.futures.army.mil/*	✓	🚫						
3	cARMY – aka MADE 2.0 *armycloud@army.mil*	✓	✓	✓	🚫				
4	AI Task Force – COES *https://coeus-il2.usgovvirginia.cloudapp.usgovcloudapi.net/*	✓	✓	✓	✓	🚫			
5	National Geospatial Intelligence Agency *https://gitlab.gs.mil/*	✓	✓	🚫	✓	✓	✓		
6	Platform One *https://repo1.dso.mil/*	✓	✓	✓	✓	✓	🚫		
7	Defense Information Systems Agency *https://gitlab.devforce.disa.mil/*	✓	✓	✓	✓	✓	+✓	✓	

SKYBORG

https://gitlab.skyborg.test.cce.af.mil/ ~ https://gitlab-76dso.cce.af.mil/

GitLab Presents an Opportunity

GitLab is a "DevOps" solution that enables individuals and groups to actively see, change, synchronize updates, and understand the evolution of (any) shared files.

FLEXIBLE OPEN FREE AVAIL. RELIABLE SECURE TENABLE FEASIBLE

1 **Naval Postgraduate School**
https://gitlab.nps.edu/

2 **Modernization Application and Data Environment (MADE)**
https://code.futures.army.mil/

3 **cARMY – aka MADE 2.0**
armycloud@army.mil

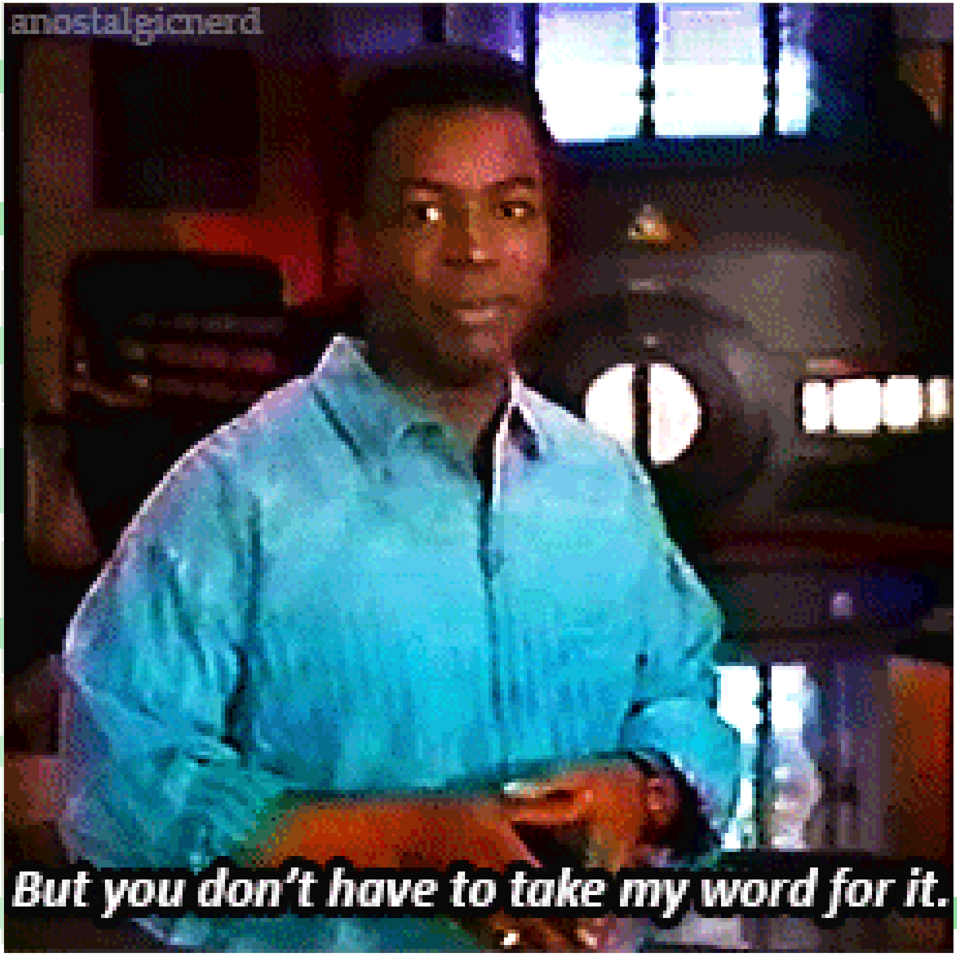
4 **AI Task Force – COES**
https://coeus-il2.usgovvirginia.cloudapp.usgovcloudapi.net/

5 **National Geospatial Intelligence Agency**
https://gitlab.gs.mil/

6 **Platform One**
https://repo1.dso.mil/

7 **Defense Information Systems Agency**
https://gitlab.devforce.disa.mil/

SKYBORG
https://gitlab.skyborg.test.cce.af.mil/ ~ https://gitlab-76dso.cce.af.mil/

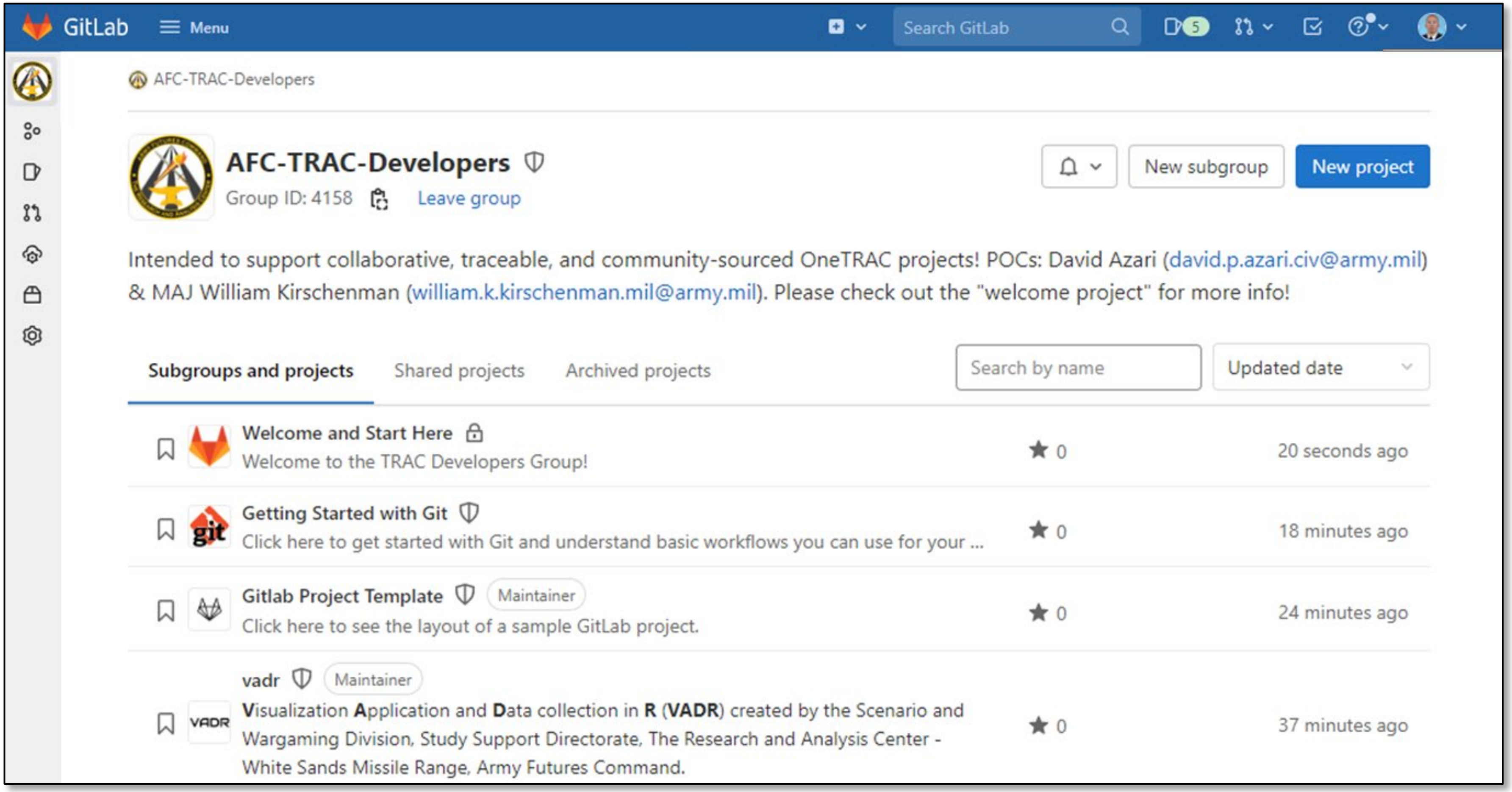


But you don't have to take my word for it.

Development Community

Landing Site and Overview

Consider this situation: You are a new analyst at TRAC, have made your way to your assigned directorate and division, and are starting the onboarding process. You are getting familiar with the work your team is doing but are unclear about what tools your team uses or could leverage from across TRAC. As part of in-processing, you received an e-mail inviting you to the “AFC TRAC Developers” group on GitLab. You proceed to the link:



AFC-TRAC-Developers

AFC-TRAC-Developers

Group ID: 4158 Leave group

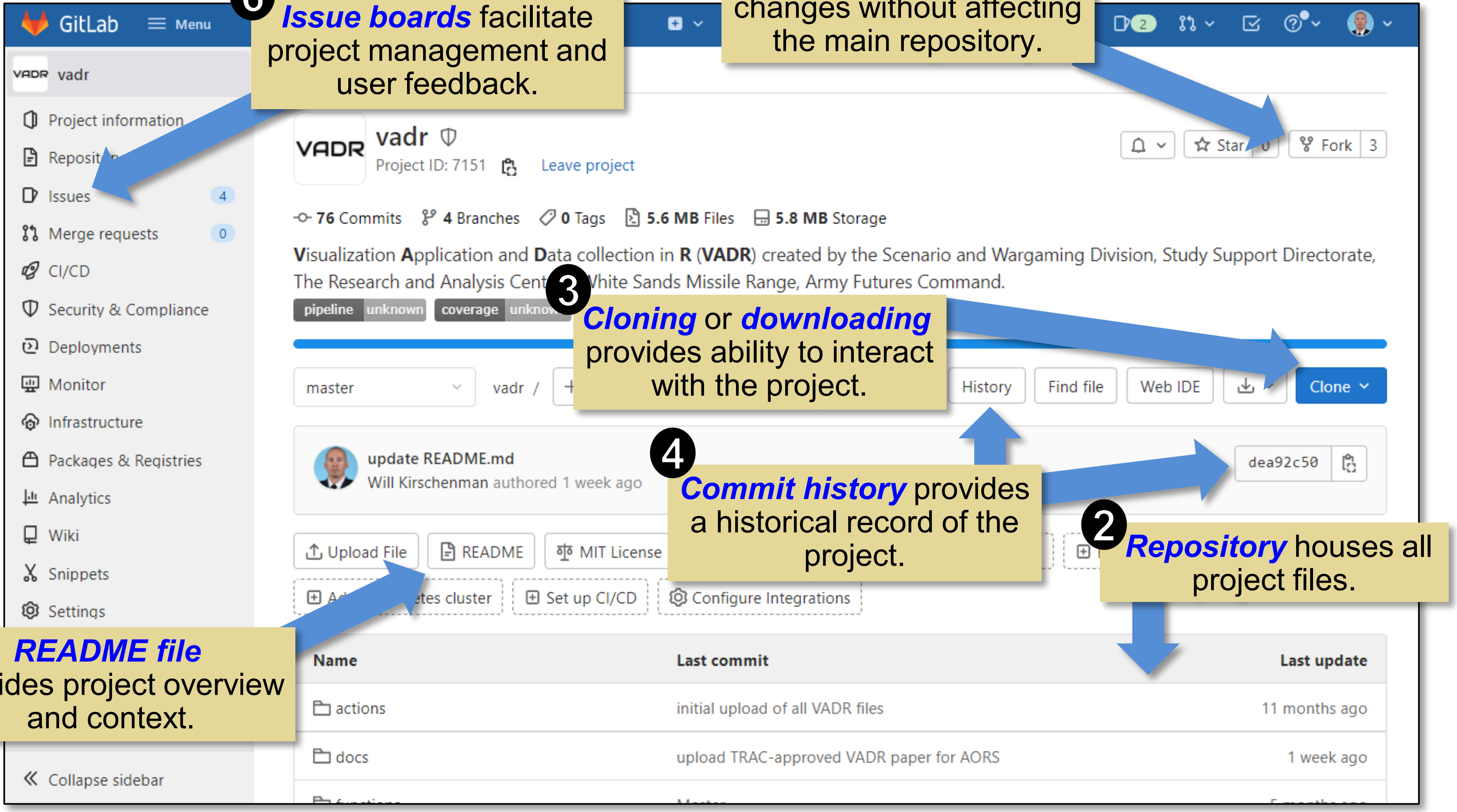
Intended to support collaborative, traceable, and community-sourced OneTRAC projects! POCs: David Azari (david.p.azari.civ@army.mil) & MAJ William Kirschenman (william.k.kirschenman.mil@army.mil). Please check out the "welcome project" for more info!

Project	Stars	Updated
Welcome and Start Here Welcome to the TRAC Developers Group!	★ 0	20 seconds ago
Getting Started with Git Click here to get started with Git and understand basic workflows you can use for your ...	★ 0	18 minutes ago
Gitlab Project Template Maintainer Click here to see the layout of a sample GitLab project.	★ 0	24 minutes ago
vadr Maintainer **V**isualization **A**pplication and **D**ata collection in **R** (**VADR**) created by the Scenario and Wargaming Division, Study Support Directorate, The Research and Analysis Center - White Sands Missile Range, Army Futures Command.	★ 0	37 minutes ago

GitLab Project Example

Project Page

(1 of 6)



GitLab ≡ Menu

vadr

Project information

Reposit…

Issues 4

Merge requests 0

CI/CD

Security & Compliance

Deployments

Monitor

Infrastructure

Packages & Registries

Analytics

Wiki

Snippets

Settings

« Collapse sidebar

VADR vadr

Project ID: 7151 Leave project

Star 0 | Fork 3

76 Commits · 4 Branches · 0 Tags · 5.6 MB Files · 5.8 MB Storage

Visualization **A**pplication and **D**ata collection in **R** (**VADR**) created by the Scenario and Wargaming Division, Study Support Directorate, The Research and Analysis Cent… …hite Sands Missile Range, Army Futures Command.

pipeline unknown | coverage unknown

master ⌄ vadr / + … History | Find file | Web IDE | Clone ⌄

update README.md
Will Kirschenman authored 1 week ago

dea92c50

Upload File | README | MIT License | …

Ad… …etes cluster | Set up CI/CD | Configure Integrations

Name	Last commit	Last update
actions	initial upload of all VADR files	11 months ago
docs	upload TRAC-approved VADR paper for AORS	1 week ago

1 ***README file*** provides project overview and context.

2 ***Repository*** houses all project files.

3 ***Cloning*** or ***downloading*** provides ability to interact with the project.

4 ***Commit history*** provides a historical record of the project.

5 ***Forking*** a project allows an analyst to make changes without affecting the main repository.

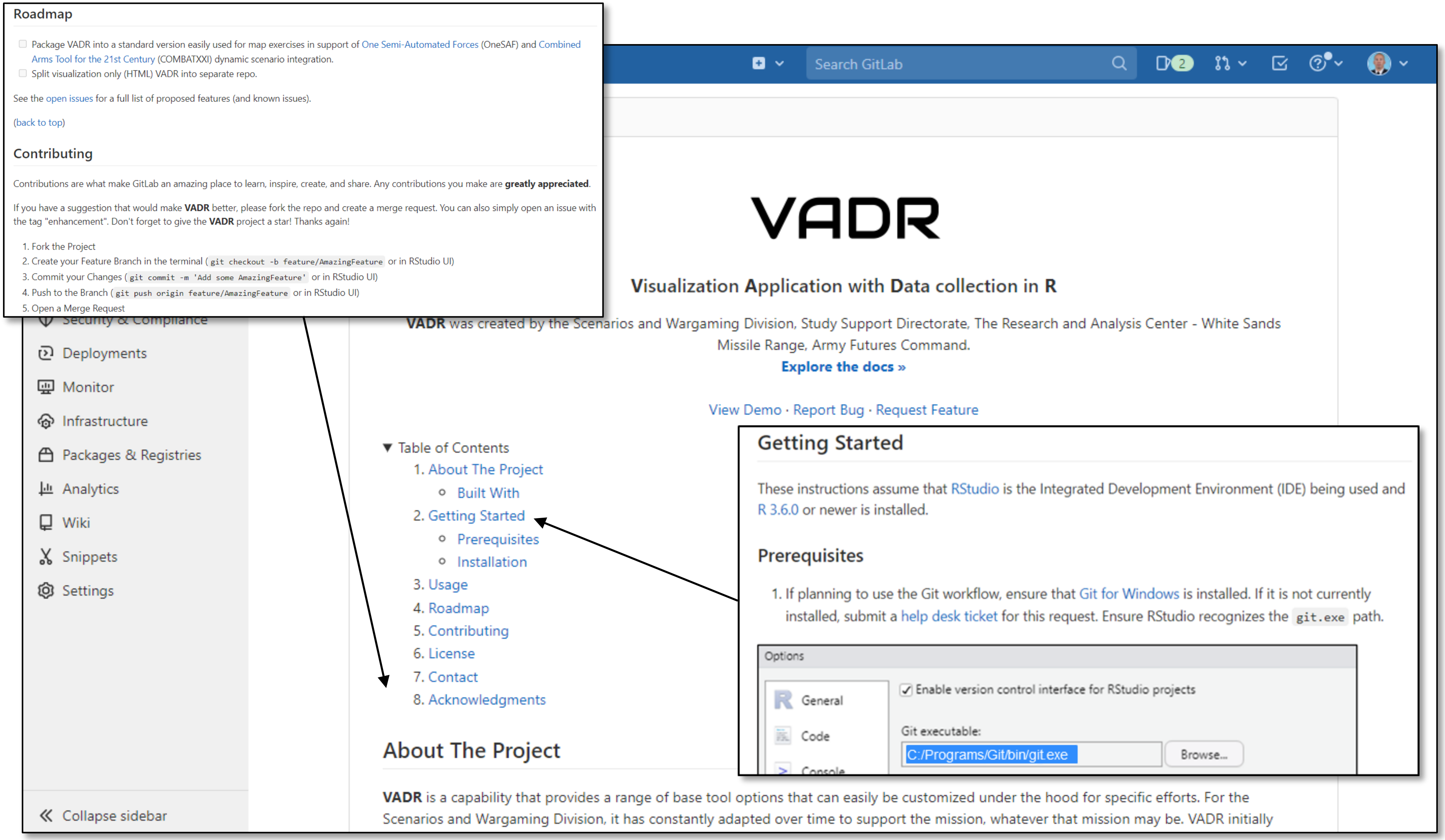
6 ***Issue boards*** facilitate project management and user feedback.

GitLab Project Example

README

(2 of 6)

The ***README*** file tells other analysts *why* the project is useful, *what* they can do with it, *how* they can use it, and *who* to contact for support and to coordinate updates.



Roadmap

- ☐ Package VADR into a standard version easily used for map exercises in support of One Semi-Automated Forces (OneSAF) and Combined Arms Tool for the 21st Century (COMBATXXI) dynamic scenario integration.
- ☐ Split visualization only (HTML) VADR into separate repo.

See the open issues for a full list of proposed features (and known issues).

(back to top)

Contributing

Contributions are what make GitLab an amazing place to learn, inspire, create, and share. Any contributions you make are **greatly appreciated**.

If you have a suggestion that would make **VADR** better, please fork the repo and create a merge request. You can also simply open an issue with the tag "enhancement". Don't forget to give the **VADR** project a star! Thanks again!

1. Fork the Project
2. Create your Feature Branch in the terminal (`git checkout -b feature/AmazingFeature` or in RStudio UI)
3. Commit your Changes (`git commit -m 'Add some AmazingFeature'` or in RStudio UI)
4. Push to the Branch (`git push origin feature/AmazingFeature` or in RStudio UI)
5. Open a Merge Request

VADR

Visualization Application with Data collection in R

VADR was created by the Scenarios and Wargaming Division, Study Support Directorate, The Research and Analysis Center - White Sands Missile Range, Army Futures Command.

Explore the docs »

View Demo · Report Bug · Request Feature

▼ Table of Contents

Getting Started

These instructions assume that RStudio is the Integrated Development Environment (IDE) being used and R 3.6.0 or newer is installed.

Prerequisites

1. If planning to use the Git workflow, ensure that Git for Windows is installed. If it is not currently installed, submit a help desk ticket for this request. Ensure RStudio recognizes the `git.exe` path.

☑ Enable version control interface for RStudio projects

Git executable: C:/Programs/Git/bin/git.exe

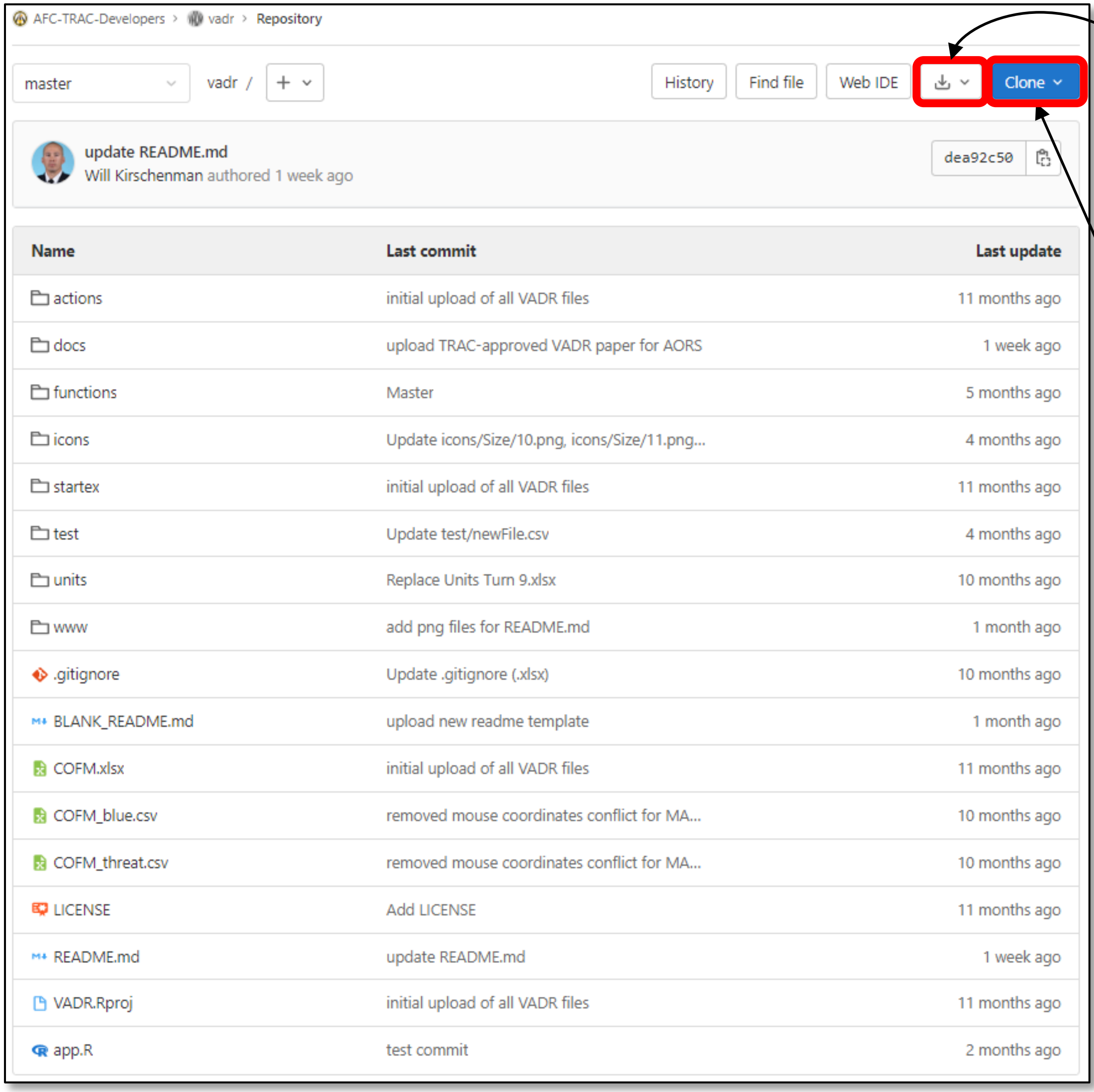
About The Project

VADR is a capability that provides a range of base tool options that can easily be customized under the hood for specific efforts. For the Scenarios and Wargaming Division, it has constantly adapted over time to support the mission, whatever that mission may be. VADR initially

GitLab Project Example

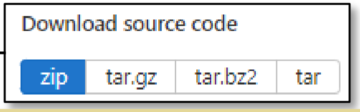
Cloning or Downloading the Repository (3 of 6)

A ***repository*** contains all of a *project's files* and each file's *revision history*. *Discussing and managing the project's work* occurs within the repository.

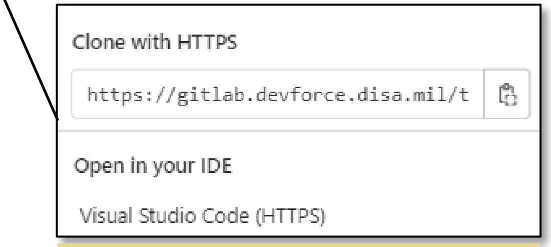


update README.md
Will Kirschenman authored 1 week ago
dea92c50

Name	Last commit	Last update
actions	initial upload of all VADR files	11 months ago
docs	upload TRAC-approved VADR paper for AORS	1 week ago
functions	Master	5 months ago
icons	Update icons/Size/10.png, icons/Size/11.png...	4 months ago
startex	initial upload of all VADR files	11 months ago
test	Update test/newFile.csv	4 months ago
units	Replace Units Turn 9.xlsx	10 months ago
www	add png files for README.md	1 month ago
.gitignore	Update .gitignore (.xlsx)	10 months ago
BLANK_README.md	upload new readme template	1 month ago
COFM.xlsx	initial upload of all VADR files	11 months ago
COFM_blue.csv	removed mouse coordinates conflict for MA...	10 months ago
COFM_threat.csv	removed mouse coordinates conflict for MA...	10 months ago
LICENSE	Add LICENSE	11 months ago
README.md	update README.md	1 week ago
VADR.Rproj	initial upload of all VADR files	11 months ago
app.R	test commit	2 months ago



Downloading a repository creates a ***local copy*** of the repository on your computer.



Clone with HTTPS
https://gitlab.devforce.disa.mil/t
Open in your IDE
Visual Studio Code (HTTPS)

Cloning a repository also allows ***synchronization between the local and remote repository locations***.

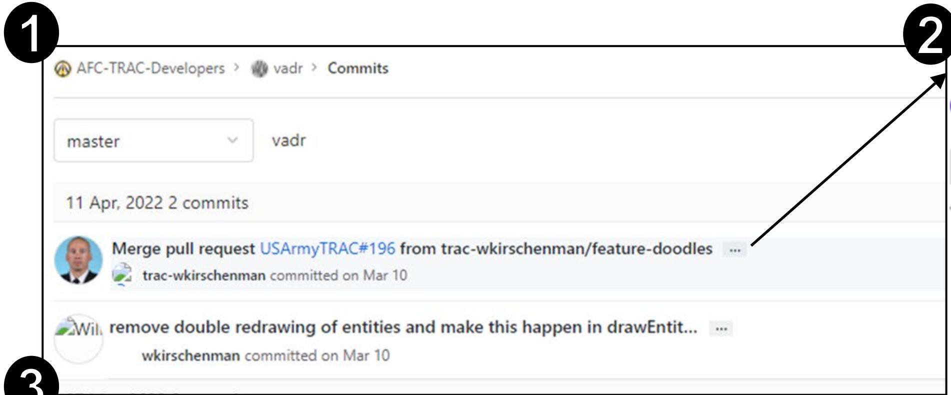
GitLab Project Example

Commit History

(4 of 6)

A ***commit*** is a *snapshot of an entire project's repository* at a specific time. Commits should be based on *logical units of change* and should *tell a story of the history* of the repository and how it came to be that way. *Merge requests* let the developer tell others about a *group of changes* pushed to a branch in the remote repository.

1

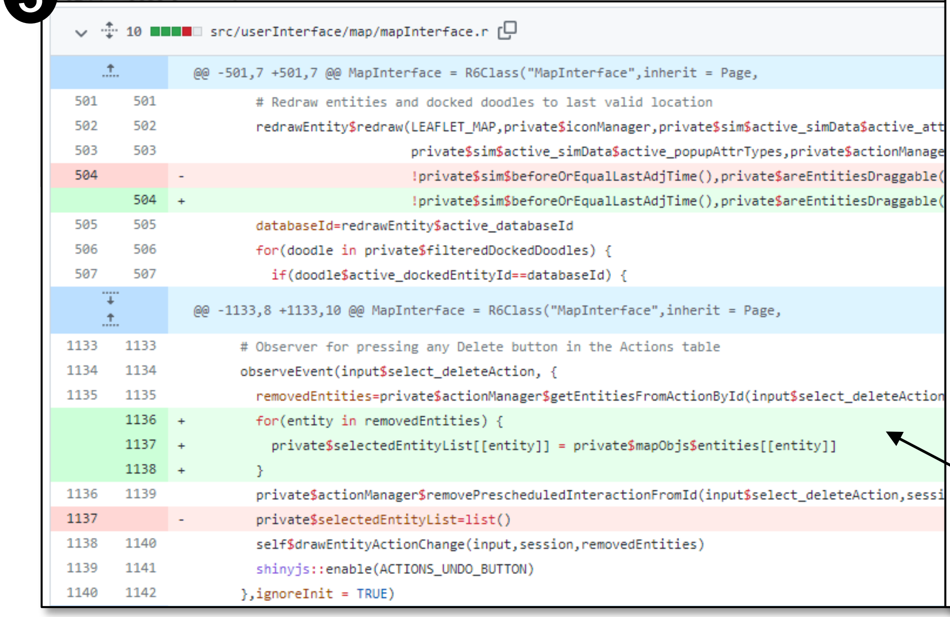


11 Apr, 2022 2 commits

Merge pull request USArmyTRAC#196 from trac-wkirschenman/feature-doodles ...
trac-wkirschenman committed on Mar 10

remove double redrawing of entities and make this happen in drawEntit... ...
wkirschenman committed on Mar 10

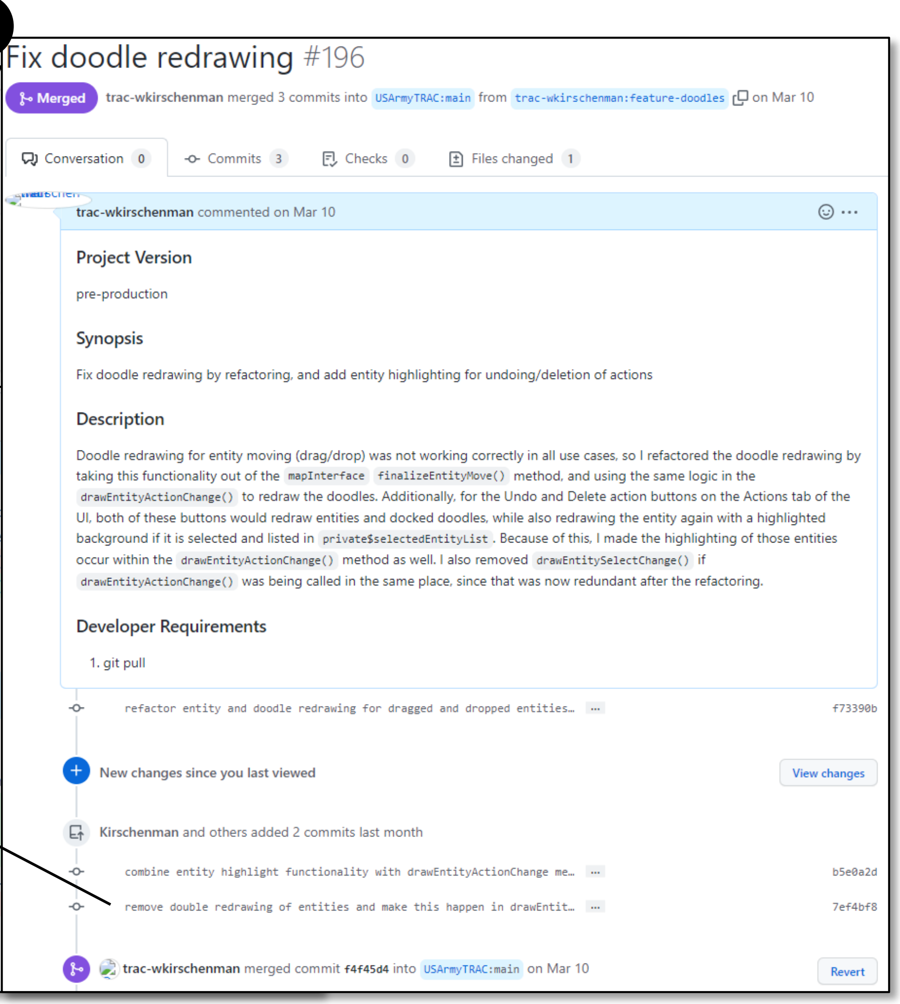
3



```
10 src/userInterface/map/mapInterface.r

      @@ -501,7 +501,7 @@ MapInterface = R6Class("MapInterface",inherit = Page,
501   501           # Redraw entities and docked doodles to last valid location
502   502           redrawEntity$redraw(LEAFLET_MAP,private$iconManager,private$sim$active_simData$active_att
503   503                                 private$sim$active_simData$active_popupAttrTypes,private$actionManage
504         -                               !private$sim$beforeOrEqualLastAdjTime(),private$areEntitiesDraggable(
      504   +                               !private$sim$beforeOrEqualLastAdjTime(),private$areEntitiesDraggable(
505   505           databaseId=redrawEntity$active_databaseId
506   506           for(doodle in private$filteredDockedDoodles) {
507   507             if(doodle$active_dockedEntityId==databaseId) {
      @@ -1133,8 +1133,10 @@ MapInterface = R6Class("MapInterface",inherit = Page,
1133  1133        # Observer for pressing any Delete button in the Actions table
1134  1134        observeEvent(input$select_deleteAction, {
1135  1135          removedEntities=private$actionManager$getEntitiesFromActionById(input$select_deleteAction
      1136  +       for(entity in removedEntities) {
      1137  +         private$selectedEntityList[[entity]] = private$mapObjs$entities[[entity]]
      1138  +       }
1136  1139          private$actionManager$removePrescheduledInteractionFromId(input$select_deleteAction,sessi
1137        -       private$selectedEntityList=list()
1138  1140          self$drawEntityActionChange(input,session,removedEntities)
1139  1141          shinyjs::enable(ACTIONS_UNDO_BUTTON)
1140  1142        },ignoreInit = TRUE)
```

2



Fix doodle redrawing #196

trac-wkirschenman merged 3 commits into USArmyTRAC:main from trac-wkirschenman:feature-doodles on Mar 10

trac-wkirschenman commented on Mar 10

Project Version

pre-production

Synopsis

Fix doodle redrawing by refactoring, and add entity highlighting for undoing/deletion of actions

Description

Doodle redrawing for entity moving (drag/drop) was not working correctly in all use cases, so I refactored the doodle redrawing by taking this functionality out of the `mapInterface` `finalizeEntityMove()` method, and using the same logic in the `drawEntityActionChange()` to redraw the doodles. Additionally, for the Undo and Delete action buttons on the Actions tab of the UI, both of these buttons would redraw entities and docked doodles, while also redrawing the entity again with a highlighted background if it is selected and listed in `private$selectedEntityList`. Because of this, I made the highlighting of those entities occur within the `drawEntityActionChange()` method as well. I also removed `drawEntitySelectChange()` if `drawEntityActionChange()` was being called in the same place, since that was now redundant after the refactoring.

Developer Requirements

1. git pull

```
refactor entity and doodle redrawing for dragged and dropped entities… ...   f73390b
```

New changes since you last viewed

Kirschenman and others added 2 commits last month

```
combine entity highlight functionality with drawEntityActionChange me… ...   b5e0a2d
remove double redrawing of entities and make this happen in drawEntit… ...   7ef4bf8
```

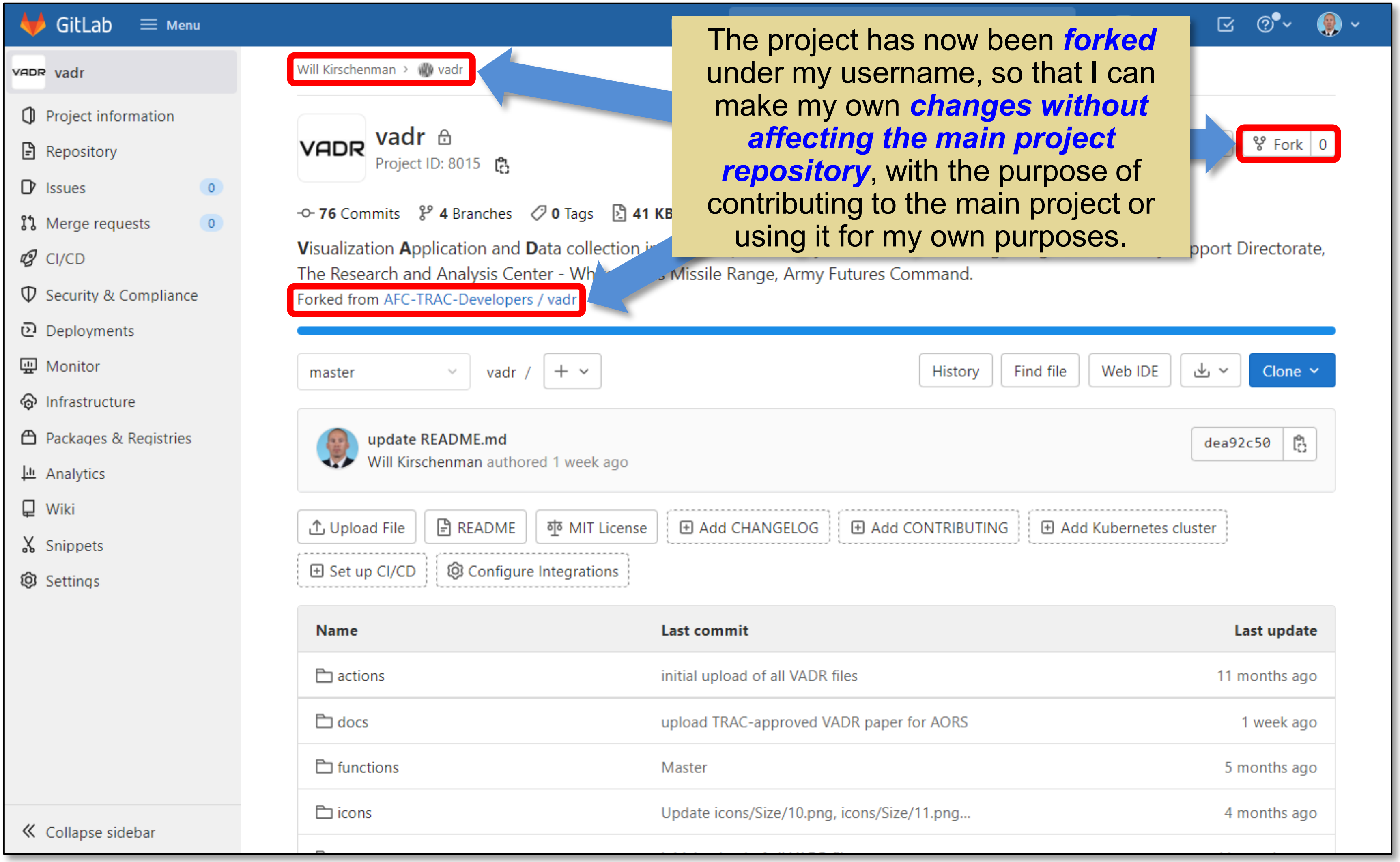
trac-wkirschenman merged commit **f4f45d4** into USArmyTRAC:main on Mar 10

GitLab Project Example

Forking

(5 of 6)

A ***fork*** is a *copy* of a project. Forking a repository allows an analyst to *make changes without affecting the original project*.



Visualization Application and Data collection i… …pport Directorate, The Research and Analysis Center - Wh… …s Missile Range, Army Futures Command.

Forked from AFC-TRAC-Developers / vadr

update README.md
Will Kirschenman authored 1 week ago

Name	Last commit	Last update
actions	initial upload of all VADR files	11 months ago
docs	upload TRAC-approved VADR paper for AORS	1 week ago
functions	Master	5 months ago
icons	Update icons/Size/10.png, icons/Size/11.png...	4 months ago

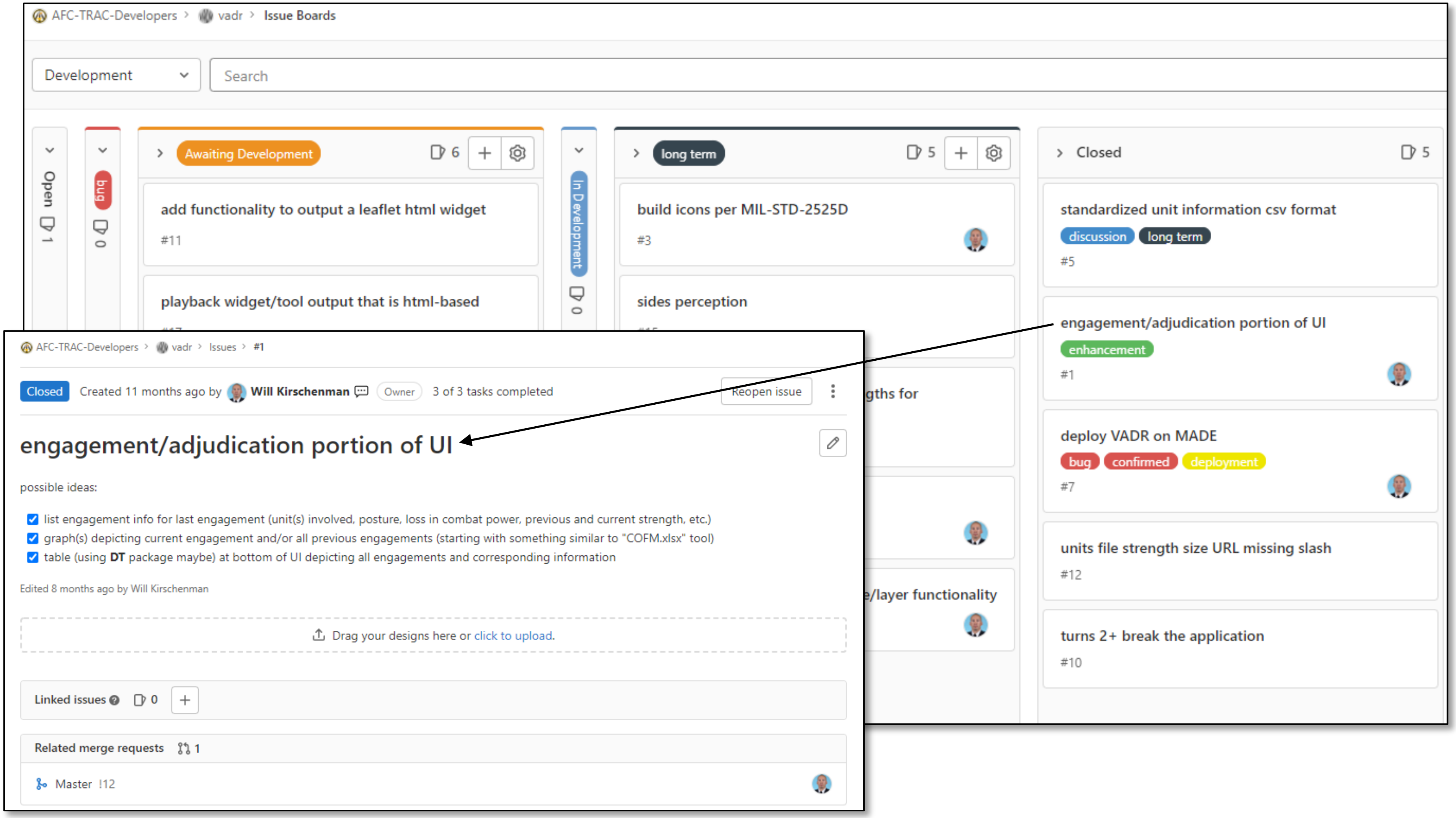
The project has now been ***forked*** under my username, so that I can make my own ***changes without affecting the main project repository***, with the purpose of contributing to the main project or using it for my own purposes.

GitLab Project Example

Issue Boards

(6 of 6)

An ***issue board*** is a *project management tool* used to plan, organize, and visualize a *workflow* for a project and its features. *Anyone in the TRAC developers group can add to the issue board*, whether as a project developer or an analyst testing features.



AFC-TRAC-Developers > vadr > Issue Boards

Development

Search

Open 1

bug 0

Awaiting Development 6
- add functionality to output a leaflet html widget #11
- playback widget/tool output that is html-based #17

In Development 0

long term 5
- build icons per MIL-STD-2525D #3
- sides perception #15
- gths for
- e/layer functionality

Closed 5
- standardized unit information csv format — discussion, long term — #5
- engagement/adjudication portion of UI — enhancement — #1
- deploy VADR on MADE — bug, confirmed, deployment — #7
- units file strength size URL missing slash #12
- turns 2+ break the application #10

AFC-TRAC-Developers > vadr > Issues > #1

Closed Created 11 months ago by Will Kirschenman Owner 3 of 3 tasks completed

Reopen issue

engagement/adjudication portion of UI

possible ideas:

- ☑ list engagement info for last engagement (unit(s) involved, posture, loss in combat power, previous and current strength, etc.)
- ☑ graph(s) depicting current engagement and/or all previous engagements (starting with something similar to "COFM.xlsx" tool)
- ☑ table (using **DT** package maybe) at bottom of UI depicting all engagements and corresponding information

Edited 8 months ago by Will Kirschenman

Drag your designs here or click to upload.

Linked issues 0

Related merge requests 1

Master !12

Summary

Advantages:

- An *open* infrastructure providing visibility to all members and awareness of what is available across directorates, centers or authority levels.
- Allows analysts to *coordinate* and *collaborate* directly.
- On-demand *reusability* of experimental, validated, and in-progress resources.
- Actionable opportunities to *"lean" on previous efforts* to inform future ones.
- Resilient, enduring, *historical record* of innovative development.
- Resonance between *technical talent* and *development opportunities* (across the community).
- Technical *project leads control permissions* to edit within broader group (default = guest).

Challenges:

- Potentially *new way of thinking / interacting* with colleagues across an organization.
- *Permission controls* may not resonate with existing authority structures.
- There is a *learning curve to the Git workflow* and building a *culture of using proper version control systems* (e.g., enforcing standards of ReadMe/description quality).
- Proper and knowledgeable administrators with *rules and procedures* in place to ensure:
  - Information in the repository meets appropriate classification levels.
  - Proper *permissions* are implemented throughout the groups, sub-groups, and projects themselves (e.g., designating guest/developer/maintainer access at any level).

Thank you for your attention!

For further information, please contact:

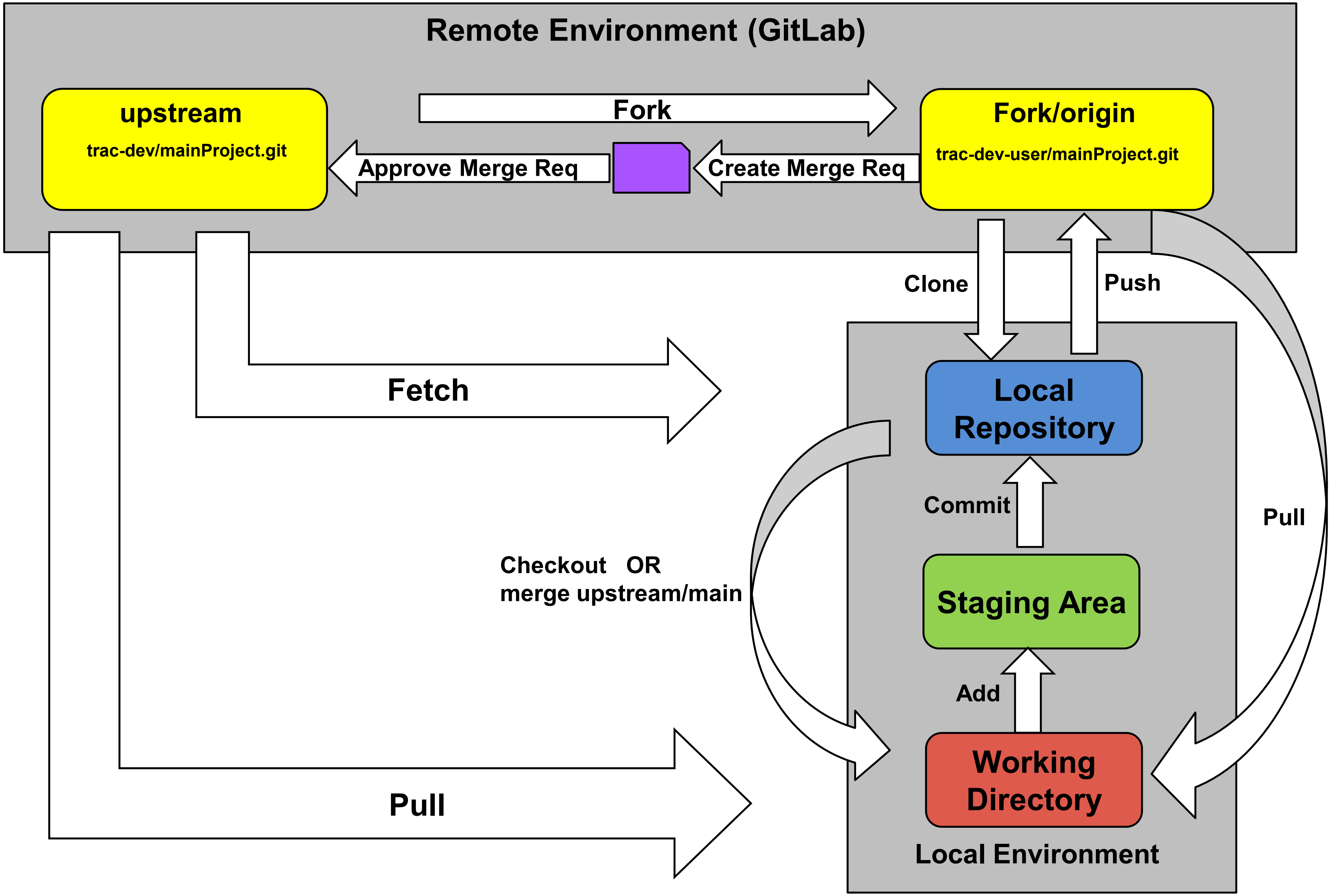
The Research and Analysis Center
BLDG 1400, MLK JR BLVD
WSMR, NM 88002

Definitions of Key Terms

- **Branch:** a line of development. A single Git repository can track an arbitrary number of branches, but your working tree is associated with just one of them (the "current" or "checked out" branch).
- **Commit (noun):** a single point in the Git history; the entire history of a project is represented as a set of interrelated commits.
- **Commit (verb):** the action of storing a new snapshot of the project's state in the Git history, by creating a new commit representing the current state of the index and advancing HEAD to point at the new commit.
- **Fast-forward:** A fast-forward is a special type of merge where you have a revision and you are "merging" another branch's changes that happen to be a descendant of what you have.
- **Fetch:** Fetching a branch means to get the branch's head from a remote repository, to find out which objects are missing from the local object database, and to get them, too.
- **Git:** a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- **GitLab:** DevOps software that combines the ability to develop, secure, and operate software in a single application using the Git distributed version control system.
- **Head:** a named reference to the commit at the tip of a branch.
- **Master:** the default development branch.
- **Merge (verb):** to bring the contents of another branch into the current branch.
- **Merge (noun):** unless it is a fast-forward, a successful merge results in the creation of a new commit representing the result of the merge, and having as parents the tips of the merged branches.
- **Origin:** the default upstream repository.
- **Porcelain:** a name for programs and program suites depending on core Git, presenting a high level access to core Git.
- **Pull:** pulling a branch means to fetch it and merge it.
- **Push:** pushing a branch means to get the branch's head from a remote repository, find out if it is an ancestor to the branch's local head, and in that case, putting all objects, which are reachable from the local head, and which are missing from the remote repository, into the remote object database, and updating the remote head. If the remote head is not an ancestor to the local head, the push fails.
- **Rebase:** to reapply a series of changes from a branch to a different base, and reset the head of that branch to the result.
- **Repository:** a collection of refs together with an object database containing all objects which are reachable from the refs, possibly accompanied by meta data from one or more porcelains.
- **Working tree:** the tree of actual checked out files.

All definitions retrieved from https://git-scm.com/docs/gitglossary

Git Fork/Pull Workflow


Remote Environment (GitLab)
upstream
trac-dev/mainProject.git
Fork
Fork/origin
trac-dev-user/mainProject.git
Approve Merge Req
Create Merge Req
Clone
Push
Fetch
Local Repository
Commit
Pull
Checkout OR
merge upstream/main
Staging Area
Add
Working Directory
Pull
Local Environment