OSPO CON @ THE LINUX FOUNDATION OPEN SOURCE SUMMIT NORTH AMERICA

# Repository Cohorts

## How OSPOs Can Programmatically Categorize All Their Repositories

Remy DeCausemaker, Justin Gosses, Natalia Luzuriaga, Isaac Milarsky, James Siri

# Live Demos of Repository Cohorts to Try During Talk or After

*Demo that takes any GitHub organization but only has partial data*
*https://aka.ms/RepositoryCohorts/Demo/YourGitHubOrg*

*Demo using Microsoft repository metadata, no data gaps:*
*https://aka.ms/RepositoryCohorts/Demo/Microsoft*

# What Are Repository Cohorts

- The concept of programmatic manipulation of raw repository metadata into higher abstract forms
- Allows for quickly understanding of the defining characteristics of a repository, its community, and its underlying usage
- Easily customizable with repeatable usage for minimal additional investment

# Repository Cohorts Definition

- *Repository cohorts are reusable, standardized, programmatically determined <u>group labels</u>.*
- *Enable OSPOs to treat repositories based on their important characteristics rather than treating every repository as the median repository*

# Repository Cohorts - In Practice

For each repository, each cohort has a value of either true or false. A continuous dimension like “age” will be split into a number of cohorts. In a group of cohorts, like age, only one cohort will be true. Other cohorts are singular like “does repo use GitHub Actions”.

| | cohort_age_ Baby 30d | cohort_age_ Toddler 30to90d | cohort_age_ Teen 90to365d | cohort_age_ Adult 365to1095d | cohort_age_ senior More1095d | Cohort_ githubActions |
|---|---|---|---|---|---|---|
| Repo A | False | False | False | **True** | False | **True** |
| Repo B | **True** | False | False | False | False | **True** |

# What Problems Do Repository Cohorts Solve?

- OSPOs sometimes have more repositories than they could realistically read and remember
- Some problems require applying rules/policies/guidance to only a subset of repositories
- Collecting and analyzing raw repository metadata values takes considerable time, so it is only done when time and expected value seems like a good investment

# How Do Repository Cohorts Help?

| Instead of… | Repository Cohorts... |
|---|---|
| ..manually reading repositories to understand what they're about | ..uses metadata to work at any scale |
| ..creating queries against raw metadata for each project | ..creates ability to identify cohorts once, then reuses in many projects |
| ..having to think about combinations of raw values | ..are easy to remember higher order representations |

*Repository cohorts improve time to insights when working with repository metadata by reducing number of steps required as well as cognitive load.*

# Example Microsoft Repository Cohort Groups

- Age
  - Cohorts based on thresholds for how long that repository has existed.
- Activity
  - Cohorts are based on amounts of activity in a repository within time periods relative to project creation and current day.
- Community
  - Cohorts that describe a repos community based on number of contributors, percentage of non-Microsoft contributors, clone amounts, and ratios of stars, forks, contributors, etc.
- Content
  - Does the repo use GitHub Actions?
  - Does the repo publish a package used internally
  - Do strings in the title or README that suggest a repo is likely a sample or demo, etc.?

# Two Microsoft Repositories, Only True Cohorts

| Repository | Type of cohort | Cohorts (true values) | Cohort Definitions |
|---|---|---|---|
| Microsoft/ typescript | Age | Senior_1095d | >1095 days since created |
| | Community | federation | Contributors > 75, ratioStarGazer vs. Contributors >2 |
| | Activity | Highlycloned | Unique-clones-count per week > 500 |
| | Activity | highExternalContrib | Non-Microsoft contributors > 25% |
| | Activity | updated30d | Updated in <30 days |
| | Content | GitHubActions | Has a GitHub Action |
| Azure-Sample/Azure-OpenAI-Docs-Samples | Age | Teen90to365days | Created >90 but <365 days ago |
| | Community | toy | Contributors <6 and stars < 100 |
| | Activity | zeroExternalContrib | 0 contributors who are not Microsoft staff |
| | Content | Sample | Has string “sample” in org or repo name |



*this was true at one point in time

# CMS Repository Cohort Pipeline

## Open Source Repository Maturity Models
- Where is our project on our Open Source Journey?

## Repository Templates
- What files are required/recommended for good repository hygiene?

## Outbound Checklists
- What steps should our project take to release the repository publicly?

## Cookiecutter
- How do we know what Maturity Model Tier our new project should be in? What files are required in that Tier?

## Repolinter Configs
- What files or information is missing from our repo?

Save us Money

Save us Time

Accountability for Contract Performance

Engine for Talent

Reduce Duplicate Work

Reduce Duplicate Costs

Reduce Security Risk

Reduce Continuity Risk

Establishing a Baseline: Repo Metrics, Maturity Models, Templates, and Checklists @ 4:10-4:50pm RM447-448
https://ossna2024.sched.com/event/1aBPh

# CMS Repository Cohort Definitions: Maturity Model Tiers

| Level | Name | Purpose | Description |
|---|---|---|---|
| **Tier 0** | **Private Repository** | Experimental, Historical | Project is **private**, usually with a single developer. Typically **working projects**, example code, and **early prototypes**. |
| **Tier 1** | **One-Time Release** | Publication for Informational, Accountability, Transparency Purposes | Project released publicly, but ***without planned future activity*** or maintenance from original author(s). |
| **Tier 2** | **Close Collaboration** | Collaboration with smaller, mostly internal teams | Project within a team or Operational Division (OpDiv), Internal Repo for ***Innersource-style work*** |
| **Tier 3** | **Working in Public** | Collaboration in the open with smaller, semi-open teams | Project developed Open Source by CMS or a CMS contractor, public website hosted on GitHub, tool or utility used in CMS official business by the public. ***Limited external contribution, CMS-led (by choice or by statute.)*** |
| **Tier 4** | **Community Governance** | Collaboration broadly in public | Project donated to or stewarded by an external community, open standard that welcomes public input, mature open source project that purposefully develops an ***open governance structure***. |

# CMS Repository Cohort Definitions: Maturity Model Tiers

Public Repo on GitHub!

https://github.com/dsacms/repo-scaffolder

## cookiecutter cmdline tool

TIER 0 TIER 1 TIER 2 TIER 3 TIER 4

DOES YOUR REPOSITORY HAVE MORE THAN 1 CONTRIBUTOR? FALSE TRUE

DO YOU PLAN ON CUTTING MORE THAN ONE RELEASE? FALSE TRUE

DO YOU PLAN ON HAVING OTHER INDIVIDUALS/TEAMS OUTSIDE THE AGENCY WORK WITH YOU? FALSE TRUE

DO YOU PLAN ON HAVING OTHER INDIVIDUALS/TEAMS OUTSIDE THE AGENCY MAINTAIN THE PROJECT WITH YOU? FALSE TRUE

DO YOU PLAN ON HAVING OTHER INDIVIDUALS/TEAMS OUTSIDE THE AGENCY PLAN THE DEVELOPMENT ROADMAP WITH YOU? TRUE

# CMS Repository Cohort Definitions: Nadia Labels

In her book "Working in Public" by Nadia Asparouhova classifies repositories into four cohorts:

- Toys
- Clubs
- Federations
- Stadiums

| | HIGH USER GROWTH | LOW USER GROWTH |
|---|---|---|
| HIGH CONTRIBUTOR GROWTH | **Federations** (e.g., Rust) | **Clubs** (e.g., Astropy) |
| LOW CONTRIBUTOR GROWTH | **Stadiums** (e.g., Babel) | **Toys** (e.g., ssh-chat) |

Various types of open source projects, classified by user and contributor growth.

https://project-types.github.io/

# CMS Repository Cohort Definitions: Nadia Labels in Augur!

```
ratio_stargazers_to_contribs = stargazers_count / unique_contributor_count

if unique_contributor_count > 75 and ratio_stargazers_to_contribs < 2:
    return "club"
elif unique_contributor_count > 75 and ratio_stargazers_to_contribs > 2 and stargazers_count > 1000:
    return "federation"
elif unique_contributor_count < 6 and stargazers_count > 100:
    return "stadium"
elif unique_contributor_count < 6 and stargazers_count < 100:
    return "toy"

#"ContribMid" is the label for repos that don't make sense in the other
#categories. Contribs > 6 and < 75
return "contribMid"
```

See Augur API Documentation for nadia_project_labeling_badge:
https://github.com/chaoss/augur/blob/main/augur/api/metrics/repo_meta.py#L205

# How CMS Creates Repository Cohorts (1/3): Repo Metrics

## Metadata Collection

When we can, we 'hack upstream' and use existing open source tools.

In this case, we are using CHAOSS' Augur project

Augur.io Ingests public repository data and history from GitHub.com

## Cohort Creation

Augur.io uses previously mentioned nadia_project_category() to categorize projects based on methodology identified by CHAOSS OSPO Metrics WG members (Shout-out Justin Gosses!)

## Cohort Reuse

DSACMS Metrics scripts pulls Augur.io API endpoint to get Project Label and Color, and uses Shields.io to create a project badge on a Repo Report

# How CMS Creates Repository Cohorts (3/3): Repolinter

https://github.com/todogroup/repolinter

Repolinter is a tool maintained by the TODOGroup for checking repositories for common open source issues, using pre-defined rulesets. This can be run stand-alone as a script, pre-commit in your IDE, or post-commit or within CI/CD systems!

✔ = Pass
✖ = Fail
⚠️ = Warn

```
✔ license-file-exists: Found file (LICENSE.md)
✔ security-file-exists: Found file (SECURITY.md)
✔ readme-file-exists: Found file (README.md)
✔ contributing-file-exists: Found file (CONTRIBUTING.md)
✔ maintainers-file-exists: Found file (MAINTAINERS.md)
✖ codeowners-file-exists: Did not find a file matching th
⚠ governance-file-exists: Did not find a file matching th
✖ community-guidelines-file-exists: Did not find a file m
✖ code-of-conduct-file-exists: Did not find a file matchi
✔ license-contains-license: Contains license (LICENSE.md)
✔ security-contains-security-and-responsible-disclosure-p
✖ readme-contains-about-the-project: Doesn't contain Abou
```

Thanks to Chan and Satwic at the Comcast OSPO, we now have **repolinter.json** configs and rules that map to each Tier of our Open Source Repository Maturity Model!

https://github.com/DSACMS/repo-scaffolder/blob/main/tier3/%7B%7Bcookiecutter.project_slug%7D%7D/repolinter.json

# Demo: CMS.gov Open Source Repository Metrics

**CMS.gov Open Source Repository Metrics**

Home | Organizations | Projects

The Centers for Medicare and Medicaid Services is comprised of many GitHub Organizations.

**CMS-Enterprise**

55 30

**CMSgov**

230 30

**DSACMS**

Digital Service at CMS

20 12

**Enterprise-CMCS**

Center for Medicaid & CHIP Services

88 30

## Report for beneficiary-fhir-data

project type club

### Repo Stats

| Stars | Forks | Issues | Watchers | Pull Requests |
|---|---|---|---|---|
| 52 | 29 | 4 | 26 | 2251 |

### Summary Table

| Metric | Latest | Previous | Diff | % Diff |
|---|---|---|---|---|
| Commits | 5056 | 5056 | 0 | 0% |
| Issues | 4 | 4 | 0 | 0% |
| Open Issues | 0 | 0 | 0 | 0% |
| Closed Issues | 4 | 4 | 0 | 0% |
| Open Pull Requests | 19 | 19 | 0 | 0% |
| Merged Pull Requests | 1808 | 1808 | 0 | 0% |
| Closed Pull Requests | 424 | 424 | 0 | 0% |
| Forks | 29 | 29 | 0 | 0% |
| Stars | 52 | 52 | 0 | 0% |
| Watchers | 26 | 26 | 0 | 0% |

https://dsacms.github.io/metrics

# How Microsoft Creates Repository Cohorts

## Metadata Collection

A centralized data team is responsible for metadata collection across all public & internal code platforms.

They make available upon internal request multiple tables of metadata for public GitHub repos as part of their Kusto cluster, a type of cloud database.

## Cohort Creation

Microsoft OSPO has an internally visible catalog of Kusto queries, including a query for generating repository cohorts that uses the collected GitHub metadata as well as an export of ClearlyDefined.io to determine which repos build packages.

## Cohort Reuse

Repository cohorts get reused as a Kusto query that gets combined with other queries as well as in PowerBI dashboards and Jupyter notebooks.

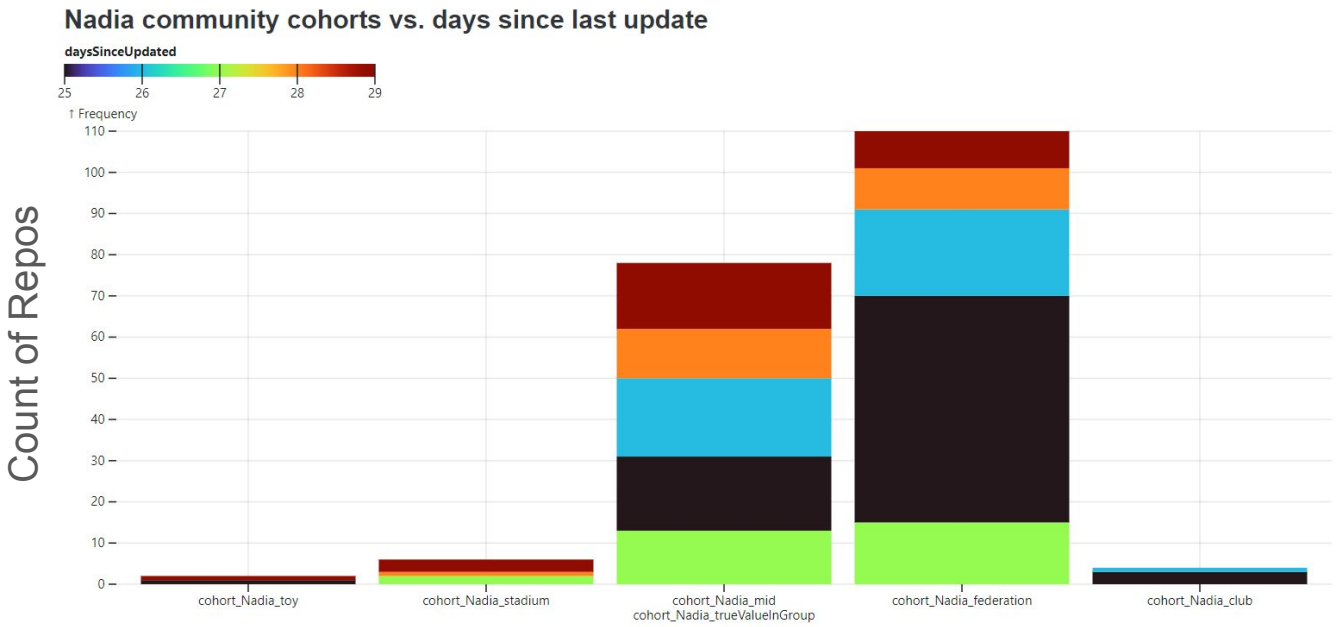
# Microsoft Examples of Using Repository Cohorts

**Demographics**: How many of each repository cohort?

**Representative sample**: Select repositories for interviews or manual investigations.

**Snapshot:** When a potential new policy only applies to a subset of repositories, we can use cohorts to produce a snapshot description of the range of activity, community shape, etc. of that subset to help design effective policy and outreach.

# Demo A: Repository Cohorts Applied to Microsoft repositories




- 10k+ repos
- No data gaps
- GitHub pages
- JavaScript



https://aka.ms/RepositoryCohorts/Demo/Microsoft

# Demo B: See Repository Cohorts for your GitHub Organization

cell 341 dataCohorts

```
SELECT full_name, owner, description, archived, stargazers_count, forks_count, open_issues_count, subscribers_count FROM dataCohorts WHERE
"forks_count" >10  LIMIT 10
```

| cohort_age_teen90to365d<br>boolean<br>all values: "false" | cohort_age_adult365to1095d<br>boolean<br>all values: "false" | cohort_age_seniorM...<br>boolean<br>all values: "true" | cohort_Nadia_missingData<br>boolean<br>all values: "true" | cohort_Nadia_mid<br>boolean<br>false true<br>2 categories | cohort_Nadia_club<br>boolean<br>all values: "false" |
|---|---|---|---|---|---|
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |
| FALSE | FALSE | TRUE | TRUE | TRUE | FALSE |

- Input your own GitHub Org and see results immediately.
- Observable page
- JavaScript
- Run live SQL queries
- Has missing fields and missing repos.

https://aka.ms/RepositoryCohorts/Demo/YourGitHubOrg

# Demo: All Cohort Definitions are in a JSON.

```
jsonThatDescribesCohortsToCreate = ▸ Array(26) [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object
// **
//  * This constant holds a JSON type data structure that describes the cohorts to be created.
//  * Each object in the array has a single key-value pair where the key is the name of the function to be called and the value is an array
//  * of arguments to be passed to the function. By calling functions in a specific order we can create key: value pairs in the data structure
//  * (aka colummns in the resulting table) that can be reused to create more complicated fields and finally cohorts. For example, "addAgeInDaysCol"
//  * is calculated and then used to calculate the "cohort_age_baby30d" cohort key/column.
// */
// export const
jsonThatDescribesCohortsToCreate = [
    //// created calculated columns used in cohorts ////
    {"addYearToRepos":[]},
    {"addAgeInDaysCol":[]},
    {"addDaysSinceCols":["updated_at","daysSinceUpdated"]},
    {"parseColumnsIntoIntegersFromStrings":["commit_stats_total_commits", "commit_stats_total_committers","commit_stats_mean_commits",
"commit_stats_dds"]},
    {"createRatioColumn":["stargazers_count","commit_stats_total_committers","ratioStargazersVsCommitters"]},
    {"createRatioColumn":["stargazers_count","forks_count","ratioStargazersVsForks"]},
    {"createRatioColumn":["subscribers_count","commit_stats_total_committers","ratioWatchersVsCommitters"]},
    //// sample or demo or example cohorts ////
    {"createCohortStringListPossibleValues":["full_name", "cohort_sample_fullName", ["sample","demo","example","tutorial"]]},
    {"createCohortStringListPossibleValues":["description", "cohort_sample_Description", ["sample","demo","example","tutorial"]]},
```

Please submit PRs

*To the cohort.js file in the Microsoft demo @*

*https://aka.ms/RepositoryCohorts/Demo/Cohort.js*

# Conclusions

- Repository Cohorts can help OSPOs deliver fit-for-purpose guidance and compliance rather than one-size-fits-all.
- Repository cohorts lower the time and cognitive burdens for using repository metadata.
- There are multiple ways to collect the raw repository metadata, each suited to your own organization. The cohort definitions are sharable and reusable.
- There are organizations and experts outside of your OSPO (shoutout to CHAOSS.community, TODOgroup.org, and others) that have developed open standards that create a shared understanding across projects, organizations, and communities.

# Live Demos of Repository Cohorts

*Demo that takes any GitHub organization but only has partial data*
*https://aka.ms/RepositoryCohorts/Demo/YourGitHubOrg*

*Demo using Microsoft repository metadata, no data gaps:*
*https://aka.ms/RepositoryCohorts/Demo/Microsoft*

*Questions & Pull Requests*
*https://aka.ms/RepositoryCohorts/Demo/Cohort.js*

# Any Repository Cohorts Questions?

We are here to answer!

opensource@microsoft.com
opensource@cms.hhs.gov

Dog Tax

THE LINUX FOUNDATION

# OPEN SOURCE SUMMIT

NORTH AMERICA