



# **Open Source EPANET: Community-Based Software Development Drinking Water Infrastructure Modeling**

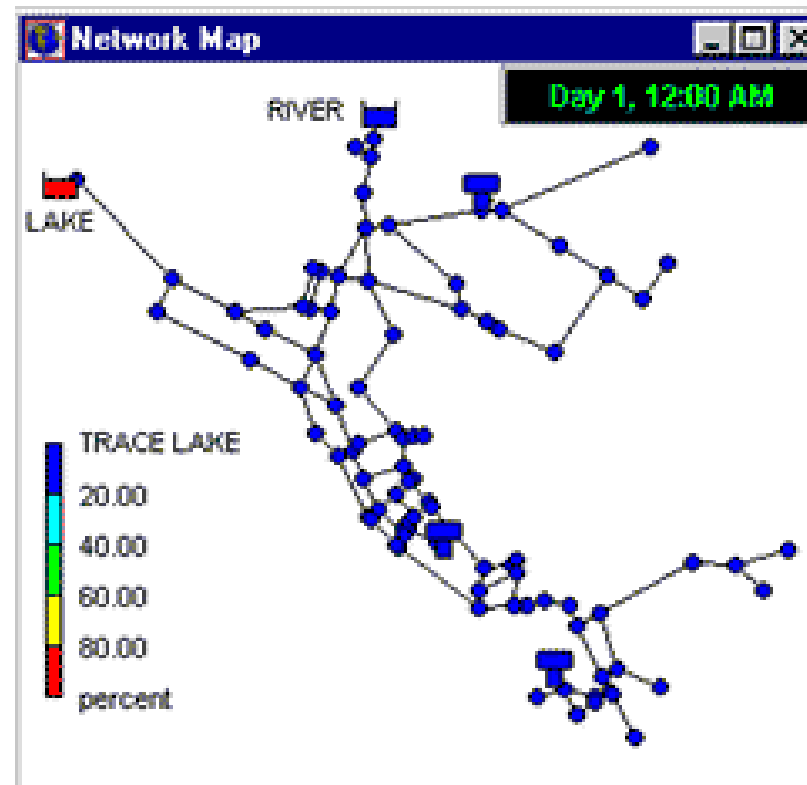
Michael Tryby, Robert Janke, Jonathan Burkhardt,  
Terra Haxton, and Regan Murray

US EPA Office of Research and Development,  
National Risk Management Research Laboratory



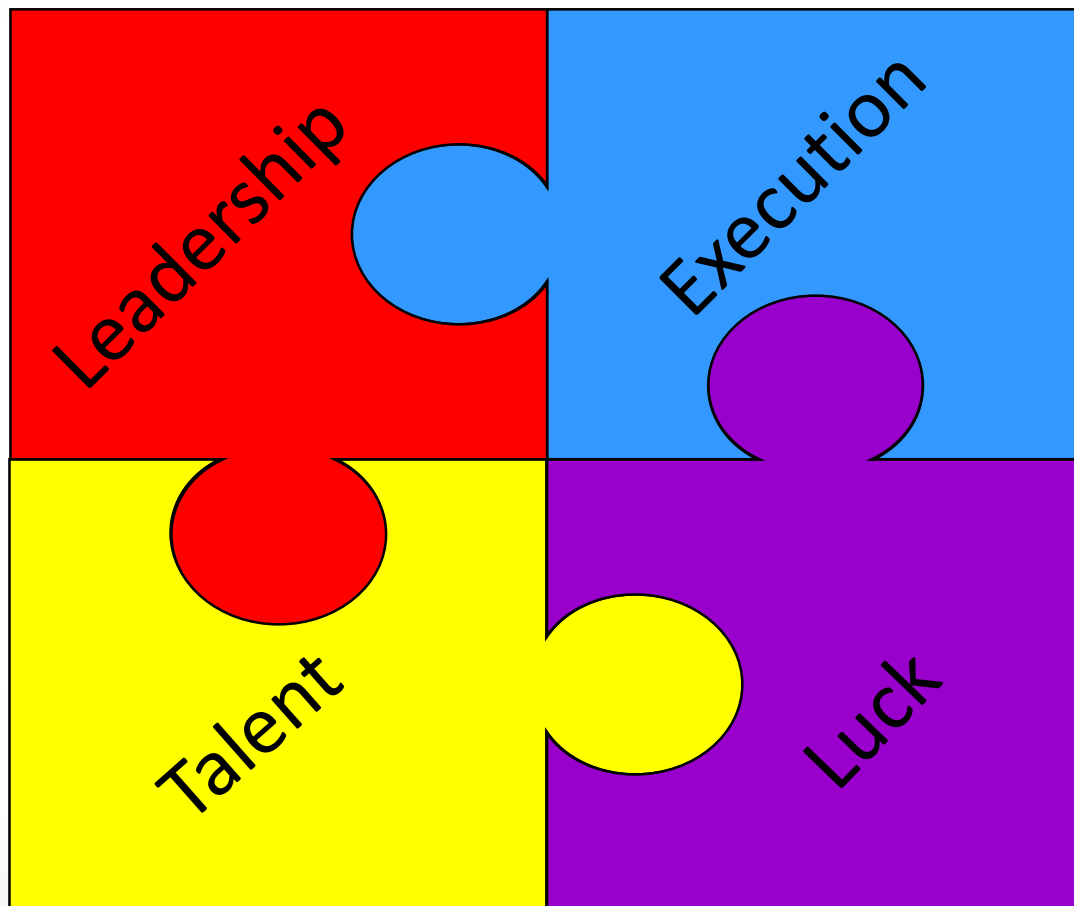


- EPANET
- Hydraulic and Water Quality Model
- Easy to use UI Application
- Programmers Toolkit
- Water Security Applications
- Third Party Applications

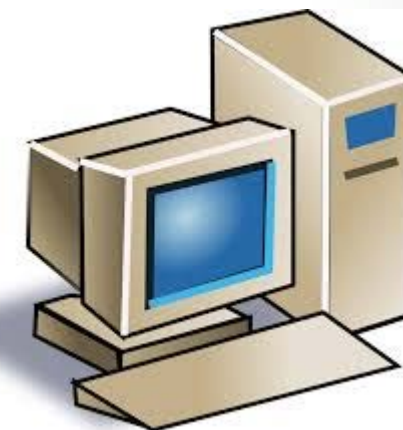




- EPA's #1 most downloaded research product
- Defacto industry standard
- Adopted by third party vendors
- Used to design \$Billions in Water Infrastructure Projects
- Integral to state and local regulatory processes
- Used to improve availability to clean water around the globe
- Used to teach fundamentals of pressurized pipe flow
- By all measures a success!



PUBLIC  
DOMAIN



- Developed from one organizations perspective
- Internal development resource constrained
- External development occurring in silos
- Slow technology transfer
- Ad hoc software QA/QC
- Long pauses in active development
- Slow adoption of new technologies
- Poor leveraging of user community
- Poor stakeholder relationships



How you think Open Source apps  
are maintained

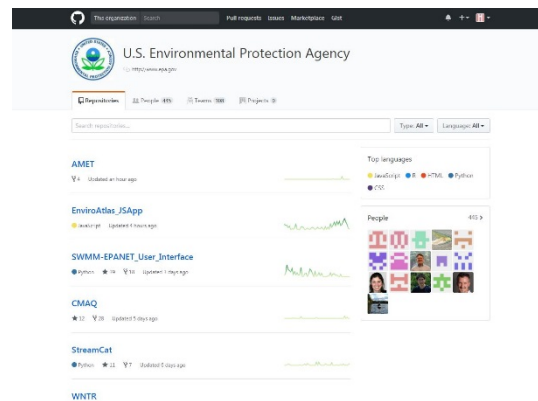


What is open source software  
development?

- Volunteer driven
- Informal
- Decentralized
- Self-organized
- Autonomous
- Difficult – like herding cats!



# Open Source EPANET: A Timeline



## Federal Source Code Policy

Discuss | Edit | View PDF of Policy

### 5.2 Participation in the Open Source Community

When agencies release custom-developed source code as OSS to the public, they should develop and release the code in a manner that (1) fosters communities around shared challenges, (2) improves the ability of the OSS community to provide feedback on, and make contributions to, the source code, and (3) encourages Federal employees and contractors to contribute back to the broader OSS community by making contributions to existing OSS projects. In furtherance of this strategy, agencies should comply with the following principles:

- A. **Language-Isolating Communities.** Whenever possible, teams releasing custom-developed code to the public as OSS should appropriately engage and coordinate with existing communities relevant to the project. Government agencies should only develop their own communities when existing communities do not satisfy their needs.
- B. **Engage in Open Development.** Software that is custom-developed for or by agencies should, to the extent possible and appropriate, be developed using open development practices. These practices provide an environment in which OSS can flourish and be repurposed. This principle, as well as the one below for releasing source code, include distributing a minimum viable product as OSS; engaging the public before official release; and drawing upon the public's knowledge to make improvements to the project.
- C. **Adopt a Regular Release Schedule.** In instances where software cannot be developed using open development practices, but is otherwise appropriate for release to the public, agencies should establish an incremental release schedule to make the source code and associated documentation available for public use.
- D. **Engage with the Community.** Similar to the requirement in the Administration's Open Data Policy, agencies should create a process to engage in two-way communication with users and contributors.

2009      2010      2012      2014      2016      2017







# 2016 - Federal Source Code Policy

## Federal Source Code Policy

[Discuss](#) | [Edit](#) | [View PDF of Policy](#)

### Section 5.2) Participation in the Open Source Community

- A) Leverage Existing Communities
- B) Engage in Open Development
- C) Adopt a Regular Release Schedule
- D) Engage with the Community
- E) Consider Community Code Contributions
- F) Provide Adequate Documentation

#### 5.2 Participation in the Open Source Community

When agencies release custom-developed source code as OSS to the public, they should develop and release the code in a manner that (1) fosters communities around shared challenges, (2) improves the ability of the OSS community to provide feedback on, and make contributions to, the source code, and (3) encourages Federal employees and contractors to contribute back to the broader OSS community by making contributions to existing OSS projects. In furtherance of this strategy, agencies should comply with the following principles:

- A. Leverage Existing Communities: Whenever possible, teams releasing custom-developed code to the public as OSS should appropriately engage and coordinate with existing communities relevant to the project. Government agencies should only develop their own communities when existing communities do not satisfy their needs.
- B. Engage in Open Development: Software that is custom-developed for or by agencies should, to the extent possible and appropriate, be developed using open development practices. These practices provide an environment in which OSS can flourish and be repurposed. This principle, as well as the one below for releasing source code, include distributing a minimum viable product as OSS; engaging the public before official release; and drawing upon the public's knowledge to make improvements to the project.
- C. Adopt a Regular Release Schedule: In instances where software cannot be developed using open development practices, but is otherwise appropriate for release to the public, agencies should establish an incremental release schedule to make the source code and associated documentation available for public use.
- D. Engage with the Community: Similar to the requirement in the Administration's Open Data Policy, agencies should create a process to engage in two-way communication with users and contributors.



How Open Source apps  
are really maintained



CommitStrip.com

- Eliminate silos
- Regularize acceptance of software contributions
- Improve tech transfer
- Stimulate community contributions
- Increase development resources
  
- Not a panacea
- I own cats!

There are many ways to contribute

- Use the software
- Help another user
- Make a feature request
- Report a bug
- Beta test
- Update / translate documentation
- Fix a bug
- Create a new feature
- Become a regular contributor





Acting as Central Consolidator:

- Contributing to EPANET dev
- Collaborating with dev partners
- Accepting contributions
- Coordinating releases
- Ensuring QA/QC

# GitHub



- EPANET @ <https://github.com/USEPA/Water-Distribution-Network-Model>
- New UI @ [https://github.com/USEPA/SWMM-EPANET\\_User\\_Interface](https://github.com/USEPA/SWMM-EPANET_User_Interface)
- OWA EPANET Version 2.1 and Version 2.2 dev
- OWA EPANET-dev (Object Oriented C++)
- QA/QC Process automation
- Python Interfaces

## New UI Application

- Maintains ease of use
- Mapping with Open Source GIS
- Plugin support
- Python scripting
- Multi-Platform
  - Windows
  - Mac OS
  - Linux







Lew Rossman  
Michael Tryby  
Sam Hatchett  
Feng Shang  
James Uber  
Tom Taxon  
Hyoungmin Woo  
Jinduan Chen  
Yunier Soad

Mike Kane  
Demetrios Eliades  
Will Furness  
Steffen Macke  
Marios Kyriakou  
Elad Salomons  
Maurizio Cingi  
Bryant McDonnell  
Andela Marchi

## Take Away Message

US EPA is

- Collaborating with multiple organizations
- Adopting Open Source and Agile Development methodologies
- Addressing development anti-patterns
- Actively working with volunteers
- Restarting core EPANET development
- Re-engineering UI Application

A graphic with the words "GET INVOLVED!" in a playful, hand-drawn style. "GET" is in dark blue, "INVOLVED" is in light blue, and the exclamation mark is in dark blue. The letters have thick outlines and a slightly 3D effect.







# Open Source Development Project

- Research project underway to investigate open source development for EPANET
- QA/QC plans (best software practices, testing, bug fixes, code documentation, Github)
- Contributor guidance (how to contribute code to EPANET)
- Licensing and waivers
- Planning for 2.2 release
- Longer term planning for 3.0 release
- Proposed session at EWRI/WDSA next May



## Open Source Timeline

IBM Mainframes	1950's
Free Software (GNU)	1980's
Linux	1990's
Open Source Initiative	1998
EPA (EPANET-MSX)	2009
Microsoft	2016
Federal Government	2016