EPIC Systems Architecture: Enabling Rapid Innovation

Kristopher Booker[1,2], Anna Kimball[1,3]

[1]NOAA, Earth Prediction Innovation Center, [2]Tomorrow.io, [3]Raytheon, Raytheon Technologies
E-mail address: kristopher.booker@noaa.gov

1. Introduction

The Earth Prediction Innovation Center (EPIC) program has enabled rapid innovation within the numerical weather model community through its implementation of various tools.

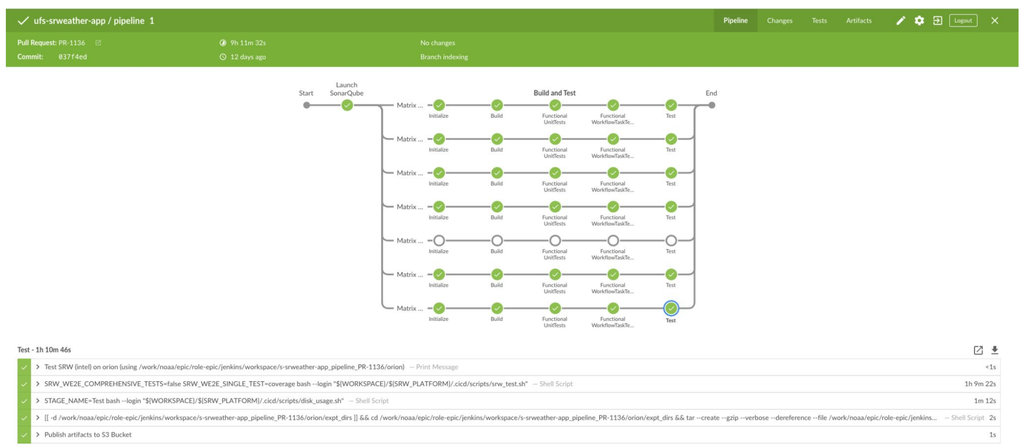
- Swift integration and feedback through CI/CD code pipelines and processes have broken down the bottlenecks that once constrained innovation.
- Public community dashboards has further progressed the level of transparency within the modeling community to foster a climate of collaboration and rapid innovation.
- A robust regression testing framework has allowed expedient testing of new scientific innovations while ensuring integrity.
- Quality control through static code analysis tools to ensure contributions conform to operational compliance standards.
- These measures lead to faster adoption of improved modeling techniques and overall forecast skill.

2. CI/CD Pipelines

GitHub Jenkins sonarqube

CI/CD Pipeline Gates

Build	Platform	Unit Test	Static Code Analysis	Regression/Functional Tests	Create Artifacts/Deploy
Checkout source code from Github repository and compile executables	Gaea Hera Jet Orion Hercules Derecho NOAA Cloud (AWS, GCP, Azure)	Run unit testing against new contributions, produce code coverage metrics against baselines	Analyze code for security vulnerabilities, bugs, code duplication, language best practices, and compliance standards.	Run regression and functional tests to ensure end-to-end compatibility	Create a package of all artifacts produced during testing and deploy the executables after completion of all quality gates



tomorrow.io

Raytheon An RTX Business

3. Public Community Dashboards and Metrics

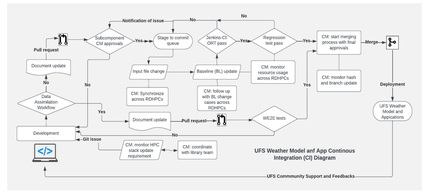
ufs-srweather-app CI/CD Artifacts

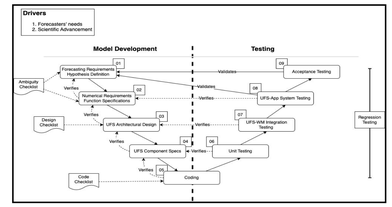
Title	State	Duration	Timestamp	Build Results	Artifacts
PR-1139 #2	RUNNING	113.42	2024-11-12T16:42:16	derecho-intel gaea-intel hera-gnu hera-intel hercules-intel jet-intel orion-intel sonarqube	Select an artifact
PR-1139 #1	FINISHED	113.42	2024-11-12T14:48:50	derecho-intel gaea-intel hera-gnu hera-intel hercules-intel jet-intel orion-intel sonarqube	Select an artifact
PR-1136 #1	FINISHED	551.54	2024-10-31T17:45:02	derecho-intel gaea-intel hera-gnu hera-intel hercules-intel jet-intel orion-intel sonarqube	Select an artifact
PR-1142 #1	FINISHED	0	2024-10-18T18:16:23	derecho-intel gaea-intel hera-gnu hera-intel hercules-intel jet-intel orion-intel sonarqube	Select an artifact

Artifact Output
Forecast skill
Software performance
Static Code Analysis
Test, build, and pipeline artifacts

- Community dashboards allow users outside the NOAA firewall to view the status of their pull requests in real time.
- Generation of build/test artifacts allows rapid analysis and resolution of any defects found in community user supplied code.
- Performance metrics allow community users to identify potential optimization or degradation by their contributions across each NOAA Tier-1 platform.

4. Testing and Static Code Analysis

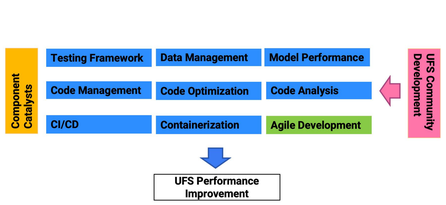


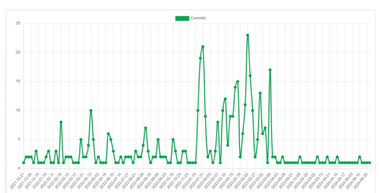


- Regression testing framework ensures new community contributions are compatible with existing application features.
- Functional and Operational Requirements Testing (ORT) ensures user contributions meet operational performance standards.
- Static code analysis ensures user community contributions are implementing industry best practices, eliminating introduction of new security vulnerabilities, and following compliance regulations.

5. Conclusions

- EPIC's systems architecture, coupled with effective code management has increased UFS community contributions.
- Increased user community contribution and collaboration with ongoing scientific improvements will lead to overall improved forecast model accuracy.





EPIC Homepage