

# EOPF-ZARR GDAL DRIVER: QUICK REFERENCE & BACKLOG PRIORITIZATION

## Visual Project Timeline

Q4 2025	Q1 2026	Q2 2026	Q3+ 2026
PHASE 1 Foundation (v0.2 MVP)	PHASE 2 Geospatial (v0.5 ARD)	PHASE 3 SAR (v1.0 Prod)	PHASE 4 Optimization (v1.x+)
✓ SLC/GRD reading	✓ GCPs	✓ Coherence	✓ S2/S3
✓ Complex64	✓ Metadata	✓ Interferog	✓ PolSAR
✓ Bursts	✓ Calibration	✓ Time-series	✓ Cloud-native
✓ GRD dual pol	✓ Multilook	✓ snap2stamps integration	✓ ARD products
✓ Version check	✓ Geocoding		
✓ CI/CD			
~8 weeks	~8 weeks	~8 weeks	ongoing
Sprint 1-4	Sprint 5-8	Sprint 9-12	Sprint 13+

## Story Priority Matrix

	EFFORT		
	Low	Medium	High
HIGH IMPACT	QUICK WINS 1.2.1-2.4	CORE CAPABILITY 1.1.1-6 2.1-2.3	DEFERRED (Plan for later)
MEDIUM IMPACT	Optional (future) 4.1.1 4.2.2	Important (Phase 2) 2.2.1 2.3.x	Schedule Phase 3+ 3.2.x
LOW IMPACT	Nice to-have	Future (v1.x+) 4.2.1	Research Only

### Key:

- **Quick Wins** (do first Sprint): Registration, metadata basics
- **Core** (Phase 1-2): Data access, geospatial intelligence
- **Deferred** (Phase 3+): Interferometry, time-series, optimization
- **Future**: Advanced processing (Sentinel-2/3, PolSAR)

## Story Dependency Graph





#### Legend:

- = Depends on
- [BLOCKED] = Waiting for external work
- [FUTURE] = Planned for later phases

## Getting Started: First 4 Sprints

### Week 1-2: Foundation Begins

#### What you need:

```
sudo apt-get install libgdal-dev cmake g++ python3-pip
git clone https://github.com/EOPF-Sample-Service/GDAL-ZARR-EOPF.git
cd GDAL-ZARR-EOPF
mkdir build && cd build
cmake .. && cmake --build . -j$(nproc)
ctest --verbose
```

**First story to tackle:** 1.1.1 (SLC Burst Reading)

#### Definition of success:

```
$ gdalinfo /path/to/eopf-slc.zarr
# Output shows burst names as bands with GDT_CComplex64 type
```

### Week 3-4: Expand Capabilities

**Next stories:** 1.1.2 (GRD), 1.1.3 (Version check), 1.1.4 (Burst select)

#### Definition of success:

```
$ gdal_translate -oo BURST=IW1_VV_001 eopf-slc.zarr burst_output.tif
$ gdalinfo eopf-grd.zarr
# Shows VV and VH as separate bands
```

### Week 5-6: Polish & Memory

**Focus:** 1.1.5 (Chunk streaming), 1.2.3 (Error handling)

#### Definition of success:

- Read 100MB burst in < 2 seconds
- Clear error messages for bad inputs
- Memory usage constant (not linear with burst size)

#### Week 7-8: Release v0.2 MVP

##### Checklist:

- ☐ All Phase 1 stories marked DONE
- ☐ Unit tests > 70% coverage
- ☐ CI/CD passing on all platforms
- ☐ README + install guide complete
- ☐ Sample notebook showing SLC → amplitude → QGIS
- ☐ Git tag v0.2 created
- ☐ Release notes published

## Sprint Velocity Projections

#### Conservative Estimate (60% velocity)

```
Sprint 1-4:    40 pts/sprint × 4 = 160 pts (Phase 1)
Sprint 5-8:    40 pts/sprint × 4 = 160 pts (Phase 2)
Sprint 9-12:   50 pts/sprint × 4 = 200 pts (Phase 3)
-----
Total:         520 pts in 12 weeks = v1.0 by Q2 2026
```

#### Optimistic Estimate (80% velocity)

```
Sprint 1-4:    50 pts/sprint × 4 = 200 pts (Phase 1 + part of P2)
Sprint 5-8:    50 pts/sprint × 4 = 200 pts (Rest of Phase 2)
Sprint 9-12:   60 pts/sprint × 4 = 240 pts (Phase 3)
-----
Total:         640 pts in 12 weeks = v1.0 by Q1 2026
```

#### Realistic (with oscillation)

```
Sprint 1:    45 pts ✓
Sprint 2:    48 pts ✓
Sprint 3:    40 pts (dip due to refactoring)
Sprint 4:    42 pts ✓
-----
Phase 1:     175 pts
Average:     44 pts/sprint
```

#### Projection for v1.0: Q2 2026 (mid-range estimate)

## Key Milestones & Gates

#### Gate 1: End of Phase 1 (Sprint 4)

**Decision Point:** Should we continue to Phase 2?

##### Go/No-Go Criteria:

- ☒ SLC/GRD reading stable
- ☒ QGIS integration working
- ☒ No critical bugs
- ☒ Team satisfied with architecture

**Go → Continue No-Go → Refactor + delay Phase 2**

#### Gate 2: End of Phase 2 (Sprint 8)

**Decision Point:** Is georeferencing & calibration solid?

##### Go/No-Go Criteria:

- ☒ GCP extraction verified

- ☒ Sigma-nought output matches SNAP
- ☒ Geocoding produces valid coordinates

Go → **Begin interferometry No-Go** → **Deep-dive on SAR domain knowledge**

### Gate 3: End of Phase 3 (Sprint 12)

**Decision Point:** Ready for production release?

**Go/No-Go Criteria:**

- ☒ Coherence/interferogram validated
- ☒ Performance benchmarks met
- ☒ Community review positive
- ☒ Documentation comprehensive

Go → **Release v1.0 No-Go** → **Focus on v1.0.1 stabilization**

## Backlog Management

### Adding New Stories (Change Request Process)

1. **Identify need:** Come from user feedback, performance data, etc.
2. **Create GitHub issue** with:
  - User story format
  - Acceptance criteria
  - Estimated points
  - Proposed sprint
3. **Refinement meeting:** Team discusses feasibility, dependencies
4. **Add to backlog:** Place in correct phase/epic
5. **Review in planning:** Prioritize against existing work

### Removing Stories (Descoping)

If a story becomes unfeasible:

1. Document reason in GitHub issue
2. Move to "Future" backlog or close
3. Update dependent stories
4. Communicate to stakeholders

### Updating Estimates

- **If story > 13 pts:** Break into smaller stories
- **If story < 1 pt:** Combine with related story
- **After Sprint completion:** Update estimates based on actual time

## Community Engagement Plan

### Touchpoints

When	Who	What
Every 2 weeks	EURAC team	Sprint planning + retro
Weekly	GitHub	PR reviews & issue updates
Monthly	Community	Webinar/discussion post
Per release	ESA/GDAL	Announce on forums

### Communication Channels

```
└─ GitHub (primary)
  └─ Issues: Bug reports, feature requests
  └─ Discussions: Questions, roadmap feedback
  └─ Projects: Sprint board (public view)
```

```
└─ EOPF Community Forum
  └─ Driver updates, integration discussions

└─ GDAL Mailing List
  └─ Driver availability, compatibility notes

└─ EURAC Website/Blog
  └─ Release announcements, tutorials
```

## FAQ & Decision Log

### Q: Why Agile/Scrum instead of Waterfall?

A: Earth observation requirements change frequently:

- New EOPF products released
- Community feedback shapes priorities
- SNAP/GDAL APIs evolve
- SAR processing best practices advance

Scrum's iterative approach handles this uncertainty.

### Q: What if Phase 2 takes longer than projected?

A: Options:

1. Extend Phase 2 to 3 sprints
2. Reduce Phase 2 scope (defer Thermal Noise to Phase 3)
3. Add team capacity (hire/contract help)
4. Re-baseline timeline to Q3 2026

### Q: How do we handle external dependencies (EOPF CPM)?

A:

- Monitor releases closely
- Maintain compatibility with last 2 CPM versions
- Document minimum requirements prominently
- Block related stories if needed
- Communicate blockers to EOPF team via GitHub

### Q: What if SNAP integrates EOPF before we finish?

A: **Positive outcome!**

- Our driver becomes complementary (not replacement)
- Users can choose SNAP or GDAL based on needs
- Collaboration opportunity with SNAP developers
- Reduces pressure on certain stories (e.g., co-registration)

## Resource Allocation Template

### Team Capacity Model

```
Lead Developer (1 FTE)
└─ 70% development
└─ 15% code review
└─ 10% documentation
└─ 5% standups/planning

Contributor 1 (1 FTE)
└─ 60% development (GIS/Zarr)
└─ 20% testing
└─ 15% documentation
└─ 5% standups

QA Lead (0.5 FTE)
└─ 50% testing/CI/CD
└─ 30% test automation
└─ 20% benchmarking

SAR Expert (0.3 FTE) [Phase 2+]
└─ 70% story estimation
└─ 20% validation testing
```

└─ 10% documentation review

Total: ~3.3 FTE equivalent

### Budget Implication (if hiring required)

Assuming €80k/year senior developer salary:

```
Phase 1 (4 sprints): ~€30k (0.5 FTE × 3 months)
Phase 2 (4 sprints): ~€40k (0.7 FTE × 3 months)
Phase 3 (4 sprints): ~€50k (1.0 FTE × 3 months)
Phase 4 (12+ sprints): ~€100k+ (ongoing)
-----
Total (Year 1):      ~€220k
```

## Success Stories & Metrics

### Measures of Success

#### By End of Phase 1:

- ☒ GitHub: 50+ stars
- ☒ Downloads: 500+ (if published to conda-forge)
- ☒ Issues: 0 critical, <5 open

#### By End of Phase 2:

- ☒ QGIS users testing
- ☒ Citation in research papers
- ☒ EOPF community endorsement

#### By End of Phase 3:

- ☒ Production pipelines using driver
- ☒ Integration with snap2stamps
- ☒ Performance matching SNAP for same operations

## Document Version Control

Version	Date	Changes
1.0	2025-12-05	Initial roadmap created
1.1	TBD	Post-Sprint 1 adjustments
1.2	TBD	Phase 2 refinement
...		

**Last Updated:** December 5, 2025

**Next Review:** End of Sprint 1

**Owner:** EURAC Earth Observation Institute