Advanced Rust (2026): Library Engineering and Testing

Lecture 4

Lukáš Hozda

Spring 2026

MFF CUNI

Library Boundaries

Public API Is A Contract

- Crate internals may change; public behavior must stay stable.
- Type names, error variants, and semantics are part of the contract.
- Document invariants, not just syntax.

Layering Strategy

- Core logic: deterministic, minimal dependencies.
- Adapter layer: I/O, environment, serialization.
- Keep side effects at edges.

```rust
pub struct Engine { /* pure core state */ }
pub struct Runner { /* io adapters */ }
```

Error Surface Design

- Avoid leaking unrelated low-level errors everywhere.
- Use domain-oriented error enums.
- Preserve source errors when useful.

```rust
#[derive(Debug)]
pub enum ParseError {
    Empty,
    InvalidNumber(String),
    Overflow,
}
```

Constructor Policy

- `new` for cheap obvious defaults.
- `try_new` for validated construction.
- Builder only when there are many optional knobs.

Feature Flags

- Optional dependencies should map to meaningful features.
- Avoid feature combinations that create incoherent APIs.
- Document feature interactions explicitly.

Testing Pyramid

- Unit tests for local invariants.
- Integration tests for public behavior.
- Property tests for algebraic expectations.
- Regression tests for known bugs.

Golden Output Tests

- Useful for formatter/CLI style libraries.
- Stable output must be intentional.
- Keep test fixtures small and reviewable.

Determinism Under Concurrency

- Public API should define ordering guarantees.
- If order is unspecified, say so explicitly.
- Deterministic tests require deterministic contracts.

Documentation As Design

- Rustdoc examples are executable tests.
- Example failure modes are as important as happy path.

```rust
/// Parses `k=v` lines.
///
/// # Errors
/// Returns `ParseError::InvalidNumber` when value is not integer.
pub fn parse_pairs(input: &str) -> Result<Vec<(String, i64)>, ParseError> {
    # unimplemented!()
}
```

SemVer Reality

- Adding enum variant can be breaking for exhaustive matches.
- Tight trait bounds can unexpectedly break downstream users.
- Test public API from an external crate perspective.

Benchmarking Protocol

- Benchmark after correctness is locked.
- Compare alternatives under representative data.
- Record compiler version and CPU in benchmark notes.

Library Review Checklist

- Contract clear?
- Error types useful?
- Tests cover edge behavior?
- Docs reflect real invariants?