

# LUASCRIPt Development Workflow

## Established by Linus Torvalds - GitHub Integration Lead

### Repository Structure

```

LUASCRIPt/
├── src/
│   ├── lexer/
│   ├── parser/
│   └── transpiler/
├── tests/
├── examples/
├── docs/
├── runtime/
├── web_ide/
└── benchmarks/
# Core transpiler implementation
# Lexical analysis
# Syntax parsing
# Code generation
# Test suite
# Example .ls files and outputs
# Documentation
# Runtime environment
# Web-based IDE
# Performance testing

```

### Branching Strategy

#### Main Branches

- `main` : Production-ready code, always stable
- `develop` : Integration branch for features
- `release/*` : Release preparation branches

#### Feature Branches

- `feature/*` : New features (e.g., `feature/enhanced-parser` )
- `bugfix/*` : Bug fixes (e.g., `bugfix/lexer-memory-leak` )
- `hotfix/*` : Critical production fixes
- `linus/*` : Linus's integration and infrastructure work

#### Team Member Branches

- `donald/*` : Donald Knuth's algorithm optimizations
- `steve/*` : Steve Jobs's UX and design improvements
- `week2/*` : Week 2 core features completion

## Development Process

### 1. Feature Development

```

git checkout develop
git pull origin develop
git checkout -b feature/your-feature-name
# Make changes
git add .
git commit -m "feat: descriptive commit message"
git push origin feature/your-feature-name
# Create PR to develop branch

```

### 2. Code Review Requirements

- **Mandatory reviewers:** Donald Knuth (algorithms), Steve Jobs (UX)

- **Linux approval required** for: Core architecture, build system, CI/CD
- **Minimum 2 approvals** for any merge to develop
- **All tests must pass** before merge

### 3. Release Process

```
git checkout develop
git checkout -b release/v1.x.x
# Final testing and bug fixes
git checkout main
git merge release/v1.x.x
git tag v1.x.x
git push origin main --tags
```

## Commit Message Standards

Following conventional commits:

- **feat:** New features
- **fix:** Bug fixes
- **docs:** Documentation changes
- **style:** Code style changes
- **refactor:** Code refactoring
- **test:** Test additions/modifications
- **chore:** Build process or auxiliary tool changes

## Testing Requirements

- **Unit tests:** All new code must have >90% coverage
- **Integration tests:** End-to-end transpilation tests
- **Performance tests:** Benchmark against previous versions
- **Manual testing:** Web IDE functionality

## Week 2 Core Features (94% Complete)

### Remaining Tasks (6%):

1. **Memory optimization** in parser (Donald's domain)
2. **Error message improvements** (Steve's UX focus)
3. **Performance benchmarking integration** (Linus's infrastructure)
4. **Documentation updates** (Team effort)

## Emergency Protocols

- **Hotfixes:** Direct to main with immediate team notification
- **Build failures:** Automatic rollback, team alert
- **Security issues:** Private branch, security team review

## Team Responsibilities

- **Linus Torvalds:** GitHub integration, CI/CD, infrastructure
  - **Donald Knuth:** Algorithm optimization, performance analysis
  - **Steve Jobs:** User experience, interface design, documentation
  - **Team:** Code review, testing, feature development
-

“Talk is cheap. Show me the code.” - Linus Torvalds