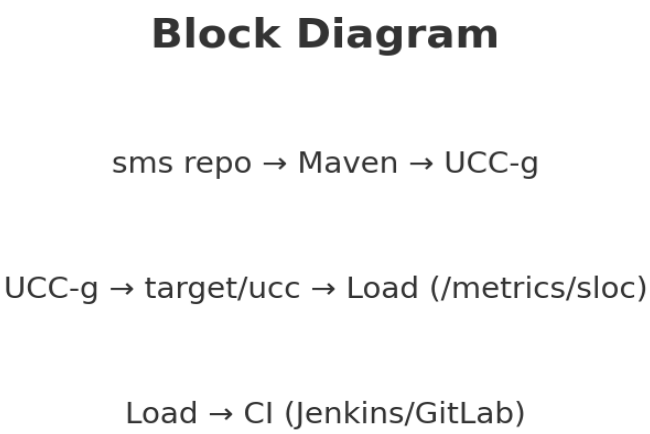
SLOC Automation — Complete Software Design Document

1. Task Description Integrate automated SLOC (Source Lines of Code) reporting into the sms build process so every produced load contains an embedded SLOC report. Objectives: - Invoke UCC-g during Maven build (prepare-package). - Capture summary and detailed per-file results. - Organize reports under /metrics/sloc/ in the load. - Archive reports as CI artifacts. - Support flexible fail policy if UCC-g is unavailable. Deliverables: - /metrics/sloc/results.csv, summary.txt, manifest.json inside each load. - CI artifacts: target/ucc/**. - Clear build logs and error handling.

2. Software Design Details

This document summarizes the software design for Milestone 1 of the SLOC Automation effort: embedding Source Lines of Code (SLOC) reports into every build load using UCC-g. The following sections describe the design details, roadmap, and UML diagrams.

Block Diagram:



Sequence Diagram:

Sequence Diagram

Developer → Maven (mvn package)

Maven → UCC-g (run at prepare-package)

UCC-g → Results → Load (/metrics/sloc) → CI

Roadmap

Roadmap Milestone 1: 1. Provision UCC-g on moncloud or Artifactory 2. Add Maven exec step at prepare-package 3. Embed results into /metrics/sloc in load 4. Add manifest.json (commit, branch, timestamp) 5. CI integration and artifact archiving 6. Validation and documentation Future Milestones: per-file per-commit tracking, filtering by component, trend dashboards.

3. Test Plan

Success Criteria: - Every load contains results.csv, summary.txt, and manifest.json. - CI artifacts include target/ucc outputs. - Excludes applied correctly. Automated Tests: - L1 Unit tests for manifest/excludes. - L2 Integration tests with stubbed ucc-g. Desktop Tests: - Run mvn package with UCC_CMD set. - Verify outputs under target/ucc and inside load. CI Pipeline Tests: - Validate presence of metrics/sloc in artifacts. Test Shot: - Conducted on moncloud build agent with feature branch.

4. Estimate of Complexity

Overall Complexity: Low → Low/Medium Drivers: - Tool availability/licensing (ucc-g). - Multi-module build consistency. - Exclusion correctness. Effort Estimate: - Baseline: 3–4 engineer-days. - With distribution hurdles: +1–2 days. Risks: - ucc-g availability and licensing. - Cross-platform differences. Confidence: 70–80% baseline, 60–70% with licensing hurdles.

5. Next Steps / Recommendations

- Confirm ucc-g distribution/licensing on moncloud. - Add exec-maven-plugin to root POM with excludes and outdir. - Update load packaging to include /metrics/sloc/**. - Add manifest.json via git-commit-id-plugin. - Configure CI agents with UCC_CMD and artifact archiving. - Validate with desktop and CI test shots. - Prepare SCR deck (pptx) and submit to Teams folder.