OSADL: Safety Critical Linux Working Group

Project # 7: DLCDM - prototype rework of DLCDM tool and extend by community interface

Rationale:

The Linux kernel development life-cycle (DLC) is not a static process but is undergoing continuous modification - in general improvements. There is considerable information on developers and commits that can be extracted from commit meta-data and evolution of this meta-data attributes of time. To allow harvesting this information for initial selection, assessment of subsystems or developers as well as process stability process impact the DLC should be systematically and continuously monitored.

The DLC evolution is captured in a database with the intent to allow for data mining on this meta-data - hence the name DLCDM. The information that can be extracted depends not only on the extraction mechanism but also on the verification of the same - from the prototype we know that there are defects (e.g. dates: Jan 1 1970, 14 Aug 2030, 25 Apr 2037…) or ambiguities (e.g. names: Peter Zijlstra, Peter Zijlstra (Intel)). While some of this can be filtered out automatically - some does need manual confirmation and some incorrect data elements can not be detected at all (e.g. if dates are within reasonable ranges but incorrect).

The DLCDM rework is not just a database but also a community verification interface to allow the data contained to be verified. Based on this data then appropriate statistical modeling can allow to extract anomalies as well as detect process level deviations.

Overview of development:

- Phase 1:	Investigation • Link to discussion thread via PASTA ? • Add reviewed-by and tested-by email/name/company • Inclusion in defconfigs • Possibility of patch series references (linking) • Upstream commit references
- Phase 2:	Cleanups of existing prototype • Review of current implementation (possibly other issues need to be addressed beyond those listed) • Name merging heuristics (developer base) with name/email update • Sanity checking (dates, name typos, general values) • Data structures (scalability issues) • Parallel execution (significant performance issues)
- Phase 3:	Requirements and design

	• Technology selection • Overall design and data design • Interface requirements • Built-in sanity checking specification • Extensions and non-interference (core functions) • Requirements and design document draft submission
- Phase 4:	Data filtering (redesign) and caching • Sublevels/sublevel ranges • Dates/date-ranges • Developers/companies • Subsystem (e.g. as rough approximation directory based) • Data caching heuristics • Data cache-merging options (investigation needed)
- Phase 5:	Open issues • Config filter (e.g. output format suitable for L2S) • Continuous update (per sublevel increment) • Merging of other tool information (coccicheck, smatch, sparse) by "tools hook" • Extension hooks (without bloating DLCDM)
- Phase 6:	Interface improvements • Text-mode/cmdline interface streamlining • GUI (community interface) • Usage statistics
- Phase 7:	Community update interface (for fixing, amending) • Registration/authentication • Change logging and role-back capabilities • Allow re-classification (e.g. non-bug/bug) • Allow name/email fixes • Configurable per-field extraction ("raw" DB)
- Phase 8:	Formal review meeting TueV and 1 day workshop (possibly with together with other tool) • Update of process specification and ATC_DLCDM • Measure and techniques adjustments (SIL2LinuxMP context) • Integration into system safety process (Annex-A/B) as well as monitoring extension and bottom up statistic model
- Phase 9:	Documentation of DLCDM in context of: • Safety related issues: competency, technology maturity. Etc.

	• Complete interpretation mapping ○ part 1 competency of developers, novelty of design/technology ○ part 3 Software operation and modification procedures, software modification and verification, functional safety assessment • Documentation of relation to overall proposed route 3_S (SIL2LinuxMP) • Resubmission of final draft to TueV
- Phase 10:	Dissemination: • Discussion: safety community, industry FLOSS community • Tool dissemination: Tool manual, HOWTO, hands-on session
Participation form:	
	• Project development and data repository • Project mailing lists (development, review) • Dissemination workshops (may be in context of other events) • Presentation in non-safety related context e.g. Linux Plumers
Responsibilities:	
	• Project legal entity: OSADL • Project resource maintenance OSADL Safety Critical Linux WG • Project Management: ????? • Technical lead: (Nicholas Mc Guire) • Implementation: (Markus Kreidl) • Non-safety related release: Nicholas Mc Guire • Certification Body: TueV Rheinland (proposed)
Effort estimation:	
	TODO
Timeline:	
Start	March 1, 2019
Milestone 1	Requirements and Design workshop
Milestone 2	Initial release version (public/community)
Milestone 3	CA Report and all related artifacts published
Maintenance phase	
Project format:	
	Partner funded project (OSADL members) - non-safety related companies participation would make sense - even if this would imply not continuing beyond

	phase 7