

First statistical analysis of Geant4 quality software metrics

E. Ronchieri¹ M. G. Pia² F. Giacomini¹

¹INFN CNAF
Bologna, Italy

²INFN
Genoa, Italy

CHEP 2015, Okinawa Japan, April 13-17, 2015

Contents

- 1 Motivation
- 2 Research Methodology
- 3 Data Analysis Methodology
- 4 A Sample of Analysis Results
- 5 Conclusions

Research challenge

Research questions

- What software metrics are valuable to measure open software used in HEP research?
- What are the recommended values for each metric?

Context

- Use of large and mature research software
- Use of software metrics tools
- Use of statistical analysis

Measuring software quality of Geant4

- Definition of a dataset of software metrics' measurements for empirical study of fault-prediction for HEP source systems
- Identification of the most effective metrics for Geant4 packages and their correlations
- Determination of metric thresholds as helpful warnings to developers on specific pieces of code

Research methodology

Described in three stages

- 1 Used existing standard, ISO/IEC 25010:2011 (former ISO/IEC 9126), to identify software characteristics for maintainability factor
- 2 Identified and evaluated software metrics tools (such as Imagix4D, SourceMonitor, LocMetrics, ACDC-Metrics, Understand) to collect a large amount of measurements of software characteristics
- 3 Exploited a set of product metrics to assess the code state

Set of Metrics

Grouped according to

- File
- Class
- Function

Categorized in

- Size
- Complexity
 - McCabe
 - Halstead
- Object-Oriented
 - Chidamber and Kemerer (CK)
- Others

Main metrics

Group	Size Metric	Source
File	Comment Ratio	Lorentz
	Declarations in File (NOD_{File})	Lorentz
	File Size (bytes)	Lorentz
	Functions in File (NOF_{File})	Lorentz
	Lines in File ($TLOC_{File}$)	Lorentz
	Lines of Source Code ($SLOC_{File}$)	Lorentz
	Lines of Comments ($CLOC_{File}$)	Lorentz
	Number of Statements (NOS_{File})	Lorentz
Variables in File (NOV_{File})	Lorentz	
Function	Lines in Function ($TLOC_F$)	Lorentz
	Lines of Source Code ($SLOC_F$)	Lorentz
	Variables in Function (NOV_F)	Lorentz

Group	Object-Oriented Metric	Source
Class	Class (LCOM)	Cohesion Chidamber and Kemerer
	Class (CBO)	Coupling Chidamber and Kemerer
	Depth of Inheritance (DIT)	Chidamber and Kemerer
	Number of Children (NOC)	Chidamber and Kemerer
	Response for Class (RFC)	Chidamber and Kemerer
	Weighted (WMC)	Methods Chidamber and Kemerer

Main metrics

Group	Complexity Metric		Source
File/Function	Intelligent (HI)	Content	Halstead
	Mental Effort (HE)		Halstead
	Program (HV)	Volume	Halstead
	Program (HD)	Difficulty	Halstead
File/Class	Average Complexity (MACC)	Cyclomatic	McCabe
	Maximum Complexity (MMCC)	Cyclomatic	McCabe
	Total Complexity (MTCC)	Cyclomatic	McCabe

Group	Complexity Metric		Source
File	Maintainability Index (MI)		Welker
	McCabe Complexity (v(G))	Cyclomatic	McCabe
	McCabe Density	Decision	McCabe
Function	McCabe Complexity (ev(G))	Essential	McCabe
	McCabe Density	Essential	McCabe

Preliminary scope of the analysis

Initial appraisals [1] concern a subset of Geant4 packages with a key role in scientific applications:

- the **geometry** package makes it possible to describe a geometrical structure and navigate through it;
- the **processes** package handles physics interactions;
- the **physics_lists** package contains physics selections.

[1] Elisabetta Ronchieri, Maria Grazia Pia, Francesco Giacomini, "Software Quality Metrics for Geant4: An Initial Assessment, " In Proceedings of the 18th Topical Meeting of the Radiation Protection & Shielding Division of ANS, RPSD 2014, Knoxville, Tennessee, USA, September 14-18, 2014.

Geant4 releases over time

Number	Release	Year
1	0.0.p04	1999
2	0.1	1999
3	1.0	1999
4	1.1	2000
5	2.0.p01	2000
6	3.0	2000
7	3.1	2001
8	3.2	2001
9	4.0.p02	2002
10	4.1.p01	2002
11	5.0.p01	2003
12	5.1.p01	2003
13	5.2.p02	2003
14	6.0.p01	2004
15	6.1	2004
16	6.2.p02	2004

Number	Release	Year
17	7.0.p01	2005
18	7.1.p01	2005
19	8.0.p01	2006
20	8.1.p02	2006
21	8.2.p01	2007
22	8.3.p02	2008
23	9.0.p02	2008
24	9.1.p03	2008
25	9.2.p04	2010
26	9.3.p02	2010
27	9.4.p04	2012
28	9.5.p02	2012
29	9.6.p04	2015
30	10.0.p04	2015
31	10.1.p01	2015

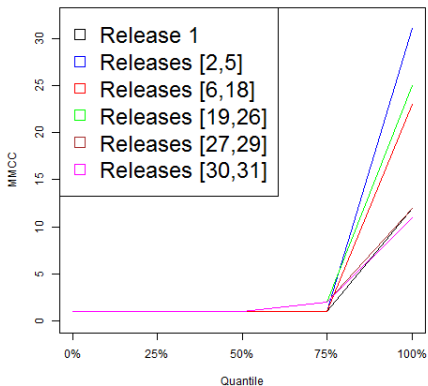
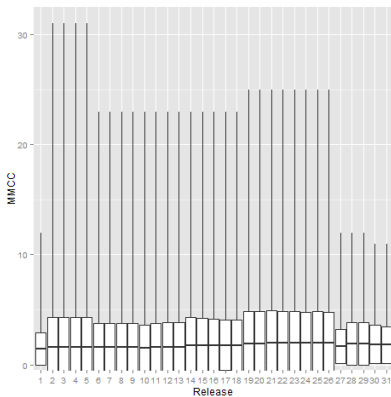
Data Analysis Methodology

Described in three stages

- 1 Use descriptive statistics for each release to get the distribution (mean and median), variance (standard deviation) and quantiles of each measure
- 2 Adopt correlations between metrics to eliminate metrics that do not provide additional insights
- 3 Identify thresholds from metrics analysis

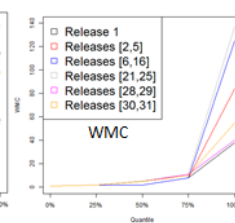
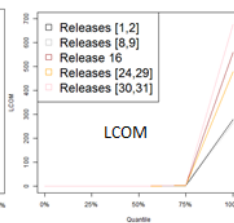
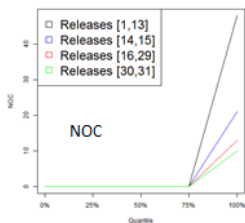
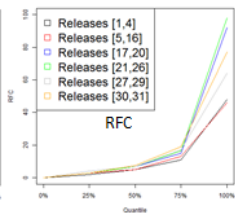
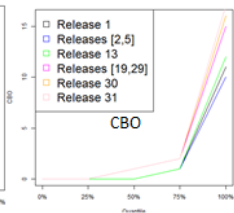
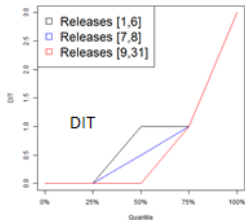
Descriptive statistics: geometry

At class group: McCabe Maximum Cyclomatic Complexity (MMCC) Metric



Descriptive statistics: geometry

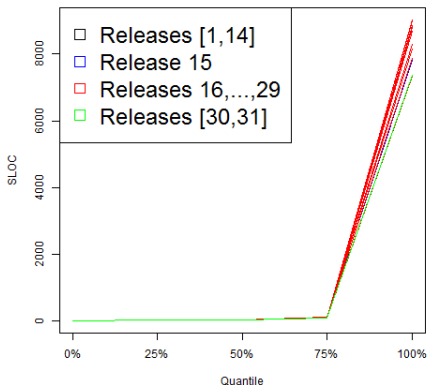
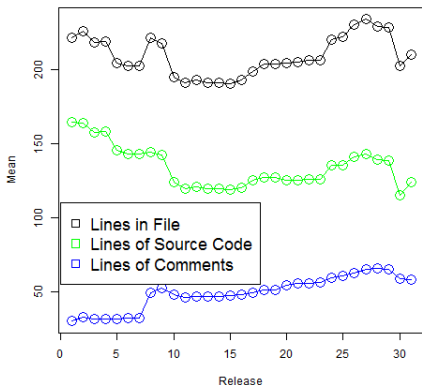
At class group: CK Object Oriented Metrics (i.e., CBO \uparrow and LCOM \uparrow in the last quartile)



Descriptive statistics: processes

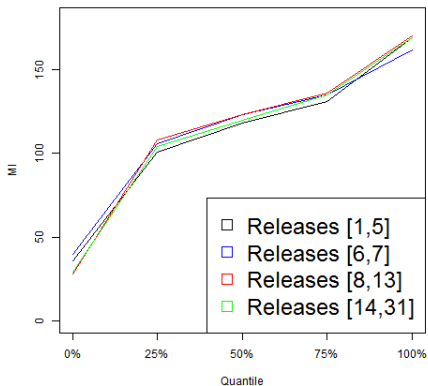
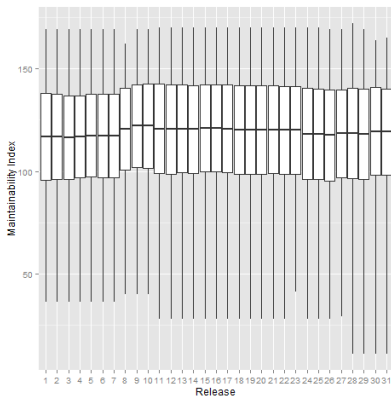
At file group: mean trend over release + SLOC

Size over Release



Descriptive statistics: processes

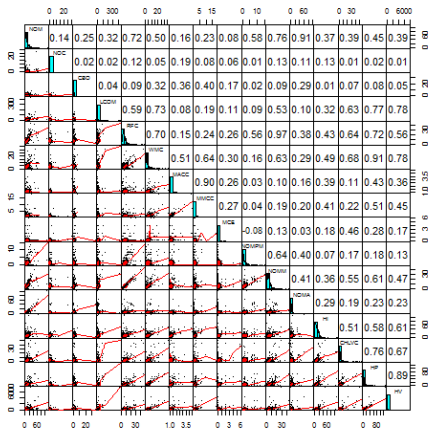
At file group: Maintainability Index (MI) Metric



Correlations between metrics: processes

At class group with Pearson r value

- the scatter graph shows metrics with a direct correlation
- choosing one metric with high correlation $0.8 \leq r < 1$ implies the others (e.g., NOM versus NOMA and RFC versus NOMM)
- medium correlation $0.5 \leq r < 0.8$ shows indirect correlation
- low correlation $0 \leq r < 0.5$ expresses very small or no correlation



Investigation on thresholds

Open Issues

- Thresholds of software goodness documented in the literature derive from specific domains, such as aerospace, telecommunication and student exercises
- They are quite old: therefore they may not reflect the evolution of the programming languages
- They cannot be blindly applied to our field

Investigation on thresholds

What thresholds should we consider?

Literature can help identifying limitations of current thresholds definition, such as those derived from experience in specific domains, error models and cluster techniques.

Research work

Identifying suitable thresholds for HEP software

Sum up

What have we done?

- Built a dataset of product metrics to investigate the quality of Geant4 software
- Identified a set of significant metrics
- Identified part of Geant4 that would benefit from close attention regarding future maintainability

In the coming months

- Provide the Geant4 developers with detailed information on the analysis of metrics
- Work on the identification of appropriate thresholds and ranges

Acknowledgements

- Gabriele Cosmo (Geant4 release manager & geometry coordinator, CERN PH/SFT) for providing prompt support during the research
- Imagix Corporation provided an extended free full license of Imagix 4D for performing this work
- Scitool provided several evaluation licenses for relevant metrics