ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
UDC/UDK 004.05:[528.91:004.9]

EVALUATION OF DESKTOP FREE/OPEN SOURCE GIS SOFTWARE BASED ON FUNCTIONAL AND NON-FUNCTIONAL CAPABILITIES

Mohammad Akbari, Mohammad Ali Rajabi

Original scientific paper

In recent years information technology has grown very rapidly. In the context of GIS, Free/Open source concept and also three-dimensional implementation have had more development too. The usage of these types of software is increasingly growing because of reasons such as security, open standards, vendor independence. In this regard, one of the most significant problems that free/open source software encounters is that these types of software do not have any specific supporter. Due to this reason the assessment and evaluation of this category of software is performed by the academic communities. On the other hand, the high rate of evolution of technology has facilitated implementation of third party software and use of three dimensions of spatial information in different processes especially in the field of GIS. Consequently, this study attempts to evaluate free/open source GIS software in the field of three dimensional applications. Therefore, after classification of Desktop free/open source GIS software, some of the selected software is evaluated based on their functional and non-functional capabilities in the context of three dimensional applications. The results of functional evaluation show that GRASS as a free/open source software competes with powerful proprietary software. Moreover, the results of non-functional evaluation show that with regard to different tests that have been done, GRASS, SAGA and ILWIS present acceptable results.

Keywords: *Desktop, FOSS, Functional analysis, GIS, non-functional analysis*

Ocjena GIS softvera radne površine besplatnog/otvorenog izvora utemeljena na funkcionalnim i nefunkcionalnim mogućnostima

Izvorni znanstveni članak

Zadnjih je godina informatička tehnologija naglo napredovala. U kontekstu GISa, koncept besplatnog/otvorenog izvora, a također i trodimenzijska primjena imali su značajan razvoj. Uporaba takvih tipova softvera sve više raste zbog razloga kao što su sigurnost, otvoreni standardi, neovisnost dobavljača. U tom smislu, jedan od najvažnijih problema s kojima se taj softver besplatnog/otvorenog izvora suočava je što takve vrste softvera nemaju nekoga tko ih posebno podržava. Zbog toga procjenu i evaluaciju takvog softvera obavljaju akademske zajednice. S druge strane, brzi razvoj tehnologije olakšao je implementaciju softvera nezavisnih proizvođača i primjenu trodimenzijske prostorne informacije u različitim procesima, naročito u području GISa. Stoga ovaj rad pokušava dati ocjenu besplatnog/otvorenog izvora GIS softvera u području trodimenzijskih aplikacija. Nakon klasifikacije besplatnog/otvorenog izvora GIS softvera radne površine, izvršena je ocjena nekih odabranih softvera na osnovu njihovih funkcionalnih i nefunkcionalnih mogućnosti u kontekstu trodimenzijskih aplikacija. Rezultati funkcionalne ocjene pokazuju da je GRASS kao softver besplatnog/otvorenog izvora konkurentan moćnom softweru na kojega se polaže vlasničko pravo. Dapače, rezultati nefunkcionalne ocjene pokazuju da su prema raznim obavljenim testovima, GRASS, SAGA i ILWIS prihvatljivi softveri.

Ključne riječi: *GIS, FOSS, funkcionalna analiza, nefunkcionalna analiza, radna površina*

1 Introduction

The trend of Free/Open Source Software (FOSS) development, especially in the field of Geospatial Information Systems (GIS), has grown rapidly in recent years. Confino and Laplante [1] said that "Open source software is widely used by government, businesses, and non-profits alike because of the financial benefits." Such progress has provided FOSS solutions to different GIS problems. Obviously, the on-going awareness of FOSS tools in GIS community helps with further expansion of these tools to new applications and solving other problems. Also as Wheeler [2] has mentioned: "Many quantitative studies have shown that, in many cases, using FOSS programs is a reasonable or even superior approach compared to their proprietary competition." However, different available FOSSs has faced some challenges to select suitable software, as Kennedy [3] has said "as with most proprietary products, OSS may not provide a solution that will satisfy everyone's requirements", therefore software evaluation is necessary.

Along with this trend towards the application of open source software goes the number of research publications that mention the use of open source software tools and libraries (see for instance [4]). Furthermore, software and algorithms developed in research projects are increasingly being published under open source licenses (e.g. [5, 6, 7, 8]).

As Steiniger and Bocher [9] explained, this rise in popularity of free GIS tools can be measured using four indicators. The first indicator is the great number of projects started in the last couple of years. As a second indicator, we see the increasing financial support by governmental organisations for the foundation of FOS GIS projects. The third indicator is the high download rates of free desktop GIS software. Finally, the fourth indicator is an increasing number of use cases of open source GIS software such as those documented by Ramsey [10] for the geospatial database PostGIS.

The rest of this paper is organized as follows. Overview of free/open source software and its trend in GIS domain is reviewed in the next section. Section 3 presents important factors for software evaluation. The necessity for free/open source GIS software evaluation is explained in Section 4. Sections 5 and 6 present functional and non-functional evaluation of desktop FOSS GIS in detail. Then we discuss our evaluation results in Section 7 and the final section concludes this study with an outlook of future works.

2 Overview of free/open source software and its trend in GIS domain

2.1 Definition

Free Software Foundation presents the following definition for Free software: "Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom for the users of the software:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this." [11]

The term "Open Source" is used by some people having the same meaning, but with a bit of difference, as "Free software". Therefore, some connect this to Free/Open Source Software or shortly FOSS. On the opposite side, there is proprietary software which is not commercial one. That is because free/open source does not mean at no charge. FOSS tools can be easily considered as commercial products too.

2.2 Users and FOSS

According to [12] users facing FOSS products can be divided into two main groups: early adopters and pragmatists. "Early adopters are comfortable using "unfinished" products, whereas pragmatists prefer to wait for the "whole"—or mature—product". Early adopters seek a competitive advantage through technology… However, this strategy comes with a price: a willingness to live with the providers of new technologies." On the other hand "Pragmatists want efficient and cost-effective products… Pragmatists expect technology to support the company's existing strategy … so they wait for a technology to become proven and then begin to implement it quickly". Regarding these different users, Golden concludes that "Changing from an early-stage company to a mature company is crossing the chasm" for free/open source trend.

Based on needs of each group of users, Golden presents a mechanism for maturity assessment of FOSS products called OSMM. He also distinguishes three different purposes of use: experimentation, pilot, production. Hence, he recommends minimum scores, coming out of OSMM, for different users and their purposes. Table 1 displays these minimum scores out of total score of 100.

2.3 Advantages versus disadvantages

Dresen [13] expresses some pros and cons for open source GI software in education as the following (Tab. 2) [13].

Table 1 Recommended minimum OSMM scores [12]

Purpose of Use	Type of User	
	Early Adopter	Pragmatist
Experimentation	25	40
Pilot	40	60
Production	60	70

Table 2 Pros and Cons of OS GI in education

Pros	Cons
* Cost of software= 0 € * Application of different GI software: diversified experience, flexibility, focus on methods * Source code is "open": new possibilities for teaching * Transparency * Distribution of software possible * High quality, scientific (adaptation of knowledge very fast) * Fast development cycle * Direct communication between user and developer	* Finally costs not= 0 € * Mostly stepwise installation * Needs of labor market * Very quick development cycle * Documentation lags behind * GUI not always intuitive * Regular update of material

Also some other advantages and disadvantages can be expressed as the following (Tab. 3).

Table 3 Advantages and disadvantages of OS GI

Advantage	Disadvantage
* Right to change and modify the source code * Easy access to the source code * Bug fixing can be immediately * Code access by many people * Free to use, free to modify and free to distribute	* Code quality and exact testing procedure * Morality of the code * Problems in utilizing some open source software as a basis for a business

3 Software evaluation: Desktop GIS

"GIS can be implemented as a comprehensive, multipurpose system (e.g. GRASS, ArcGIS), as a specialized, application oriented tool (e.g. GeoServer), or as a subsystem of a larger software package supporting handling of geospatial data needed in its applications (e.g. hydrologic modelling system, geostatistical analysis software, or a real estate services Web site). The multipurpose systems are often built from smaller components or modules which can be used independently in application oriented systems" [14].

It is clear that in using desktop GIS software, its purpose and basis should be considered. Table 4 lists factors and indicators used in this study. It is worth noting that different approaches of software in this class make a comprehensive and consistent comparison very hard. On the other hand, wide area of spatial analysis performing is a great barrier in performing a more precise analysis.

Tab. 5 presents the result of comparing available software in this category. It is noteworthy that for some factors a 1 to 5 range is applied where 5 shows the most complete implementation of that factor by the software.

4 The necessity for free/open source GIS software evaluation

Sanchez et al. [15] have presented a primarily version of web based multimedia GIS for using in international health care (between U.S. and Mexico). The design and implementation of this GIS is based on Free/Open source GIS software and standards. They emphasise that: "according to our experience with using free/open source software, and based on our responsibilities, we should mention that there is a shortage of knowledge and information in using this software. According to them, there is a need for a strong qualitative and quantitative evaluation of FOSS in general and in particular free/open source GIS software".

In addition to this, there are some other reasons that justify FOSS GIS evaluation. One of them is because usually FOSS does not have a specifically responsible person; therefore no one is responsible for its function. Also, the result of these assessments can help developers to plan software policy development and users to select software based on their needs.

In this research, FOSS GIS evaluation is done in two directions. First, some of the FOSS GIS are assessed based on their functional capabilities. Second, the non-functional capabilities of this set are evaluated.

5 Functional evaluation of desktop FOSS GIS

In software engineering, the functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. System requirements identify the process that system has to do. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionalities that show how a use case is to be fulfilled.

They are supported by non-functional requirements, which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular behaviours of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements define system application architecture, whereas non-functional requirements define technical architecture of a system.

Table 4 Factors used in analysis of Desktop GIS systems

Factor		Description
Data access (Read/Write)	Raster	DEM formats (USGS SDTS, …), Image formats (PNG, JPEG, …), GeoTIFF, ArcInfo Grid, ERMapper, HDF, MrSID, …
	Vector	Shapefile, ArcInfo Coverage, WKT, GML, KML, DGN, DXF, TIGER, CSV, …
	Table	CSV, dbf, text file
	Database	PostGIS, Oracle Spatial, ArcSDE, GeoDatabase, MySQL, SQLite, Informaix DataBlade, DB2, ODBC, …
Vector Manipulation	Create/Edit	Create (point, line, polygon, other shapes), Edit (point, line, polygon, surface), Edit capabilities (snap, cut, attribute calculator, …), Digitizing, Network editing, …
	Basic tools	Selection modes, Layout setting and Print, Attribute query (level of complexity), Spatial query (level of complexity), Basic analysis (distance, area, volume, buffer, …),
	Advanced 2D analysis	Generalization, Union, Intersect, Difference, Dissolve , Voronoi diagrams, Network analysis (best path, watershed, flow accumulation, …), Rasterization, …
	3D Analysis	TIN (Triangulation, slope, aspect, contour, line of sight, …), …
	Topology support	-
Raster analysis	2D	Raster calculator (=Raster math), Reclassification, Buffer, Gridding, Interpolation(different methods), Image processing (Filter, supervised & unsupervised classification, …), DEM analysis, Vectorization, …
	3D	Voxel Creation/Conversion(from 3D vector, to 2d raster, …), 3D raster calculator(: based on voxel), Voxel Interpolation, ,
Visualization	2D	2D view, 2D Navigation(pan, zoom, bookmark, layer tree, …)
	3D	3D view, 3D Navigation (3D zoom, 3D pan, ..), Overlay multiple layers (raster & vector), Settings (Angle, place of camera, ..), special effects (shading, …), animation creation, …
Appearance	Styling	Display change (legend, symbology, scale bar, grid)
	Labelling	Level of complexity
Temporal analysis		-
Modular (Plugin able)		-
Standards		OGC, ISO, …
Modelling and Simulations		Whether tools are aimed towards specific modelling and simulations

Functional analysis in system engineering, software engineering and commercial process engineering has been done as part of designing procedure. Typically designing procedures include definition, requirements analysis, functional analysis, resource or physical definition, non-functional analysis and performance.

Table 5 The result of comparing available software in category of Desktop GIS systems

		GRASS	QGIS	uDig	JUMP/OpenJUMP	gvSIG	SAGA	LandSerf	ILWIS	SPRING	MicroDEM
Data access (Read/Write)	Raster	√	√	√	√	√	√	√	√	√	√
	Vector	√	√	√	√	√	√	√	√	√	√
	Table	√	√	-	√	√	√	-	√	√	√
	Database	√	√	√	√	√	-	-	-	-	-
Vector manipulation	Create, Edit	√	√	√	√	√	√	√	√	√	√
	Basic tools	√	√	√	√	√	√	√	√	√	√
	Adv. 2D analysis	√	√	√	√	√	√	-	√	√	√
	3D Analysis	√	√	-	-	√	√	-	-	-	√
	Topology support	√	√	-	√	-	-	-	-	√	-
Raster analysis	2D	√	√	-	-	√	√	√	√	√	√
	3D	√	√	-	-	√	√	-	√	-	-
Visualization	2D	√	√	√	√	√	√	√	√	√	√
	3D	√	-	-	-	√	√	√	√	√	√
Appearance	Styling	√	√	√	√	√	√	√	√	√	√
	Labelling	√	√	√	√	√	√	√	-	√	√
Temporal analysis		√	√	-	-	-	-	-	-	-	-
Modular (Plugin able)		√	√	√	√	√	√		√	√	-
Standards		WMS,WFS, SFS, GML, KML	WMS,WFS,SFS, GML, KML	WMS,WFS,GML, SLD,SFS, FE	WMS,WFS,GML, SFS	WMS, WFS, WCS, CS.GML, WMC, SLD, FE	-	-	-	-	-
Modelling and Simulations		Hydrologic, erosion and pollutant, fire speed	Like GRASS	-	-	-	Cellular Automata, Fire Spreading Analysis, Hydrology, Modelling the Human Impact on Nature	-	Hydrologic	Environment modelling, Cadastre	-

5.1 Functional results

This section deals with the functional analysis of software. For functional assessment, the emphasis is on the 3D capabilities of software. Therefore, it is done with three different capabilities: Import/Export 3D data, 3D Analysis and 3D visualization and representation. The results of these evaluation and classification have been represented in Tabs. 6, 7, 8, respectively.

In the case of Import/Export of 3D data, as results show SAGA is more capable than the other two software packages. After SAGA, GRASS and ILWIS have the second and third ranks respectively.

In 3D analysis evaluation, GRASS has provided a vast range of functionalities. After GRASS, SAGA and ILWIS are the next functional capable software.

In the field of 3D visualization and representation evaluation, SAGA has a better situation than the other two software packages. After that, it is GRASS which has more capabilities. The third rank belongs to ILWIS in the case of 3D visualization and representation capabilities.

Finally, from the result of functional evaluation of this software it can be concluded that SAGA and GRASS have an acceptable condition in comparison to the proprietary GIS software. But ILWIS is not as mature as the other two software.

6 Non-functional evaluation of desktop FOSS GIS

Important responsibilities of system developers are identification of application requirements, developing software that implements requirements and assigning proper resources (processors and connecting networks). In other words, it is not enough to comply with just functional requirements. Generally, significant and vital systems have to provide security, safety, reliability, performance and the other similar requirements.

Table 6 Functional capabilities for 3D data import, export

	GRASS 6.3.0	SAGA 2.0.0	ILWIS 3.4
3D Import	* ASCII 3D: Converts a 3D ASCII raster text file into a 3D raster map layer * Vis 5D: import of 3-dimentional Vis5D files	* Load TIN * Import grids using GDAL * Import GPS data * Import ESRI E00 file * Import ESRI ArcInfo grid * Import SRTM30 DEM * Import Surfer grid * Import USGS SRTM grid * Import GStat shapes * Import shapes from XYZ	* Contour map digitizing * Importing maps from the other package
3D Export	* ASCII 3D: converts a 3D raster map layer into an ASCII text file * Vis5D: export of GRASS 3D raster map to 3D Vis5D file. * VTK: converts 3D raster maps (G3D) into the VTK-ASCII format.	* Export grids using GDAL * Export ESRI ArcInfo grid * Export grid to XYZ * Export Surfer grid * Export GStat shapes * Export shapes to generate *Export shapes to XYZ	* Exporting map to cartographic software * Export ArcInfo ASCII * Export ArcInfo E00 file

Table 7 Functional capabilities for 3D Analysis

GRASS 6.3.0	SAGA 2.0.0	ILWIS 3.4
* Conversions: - Raster series to volume - Raster 2.5D to volume - Vector to volume - Volume to raster series * Terrain Analysis: - Cumulative movement costs - Cost surface - Least cost route or flow - Shaded relief - Slope and Aspect - Terrain parameters - Textural features - Visibility * Hydrologic Modelling: - Watershed analysis * Generate Surfaces: - Fractal surfaces - Gaussian kernel density surface - Gaussian deviates surface - Plane - Random deviates surface - Random surface with spatial dependence * Contour Lines: produces a vector map layer of specified contours from a raster map layer. * Interpolate Surfaces: - Bilinear from raster points - Bilinear and bicubic from vector points - IDW from raster points - IDW from vector points - Raster contours - Regularized spline tension - Fill Null cells * Grid 3D volumes: - Manage nulls for grid 3D volume - Manage timestamp for grid 3D volume * 3D Mask:	* Geostatistics: - Kriging - Kriging with variogram fit * Slope and Aspect * Grid: - Contour lines from grid * Hydrology: - Overland flow - Soil moisture content - TOPMODEL - Water retention capacity * TIN: 1. Conversion: - Grid to TIN - Shapes to TIN - TIN to shapes 2. Terrain Analysis: - Flow accumulation - Gradient * Terrain Analysis: 1. Channels: - Channel network - D8 flow analysis - Overland flow distance to channel network - Strahler order - Vertical distance to channel network - Watershed basins 2. Lighting: - Analytical hill shading - Incoming solar radiation - Insolation - Sky view factor - Visibility 3. Morphometry: - Curvature classification - Surface specific points 4. Profiles: - Cross profiles - Cross section - Flow path profile - Profile - Profile from points - Profile from lines * Slope, aspect, and gradient grid analysis. * Creation of triangulated irregular networks (TINs) and digital terrain models (DTMs),	* Interpolation: - Density raster map - Kriging from raster - Contour interpolation - Point interpolation * DEM hydro-processing: - DEM visualization - Flow determination - Flow modification - Network and catchment extraction - Compound parameter extraction - Statistical parameter extraction * Slope, aspect, and gradient grid analysis. * Relief shading * Includes support for constrained Delaunay triangulation, allowing TINs to be created from a collection of surveyed elevations, with optional break lines and bounding areas * provides line-of-sight

establishes the current working 3D raster mask. * 3D Map Calculator: Calculates new grid 3D volume * Cross Section from Volume: * Ground flow model * Interpolate volume from vector points * Slope, aspect, and gradient grid analysis. * Creation of triangulated irregular networks (TINs) and digital terrain models (DTMs), * Relief shading * Includes support for constrained Delaunay triangulation, allowing TINs to be created from a collection of surveyed elevations, with optional breaklines and bounding areas * provides line-of-sight	* Relief shading * Includes support for constrained Delaunay triangulation, allowing TINs to be created from a collection of surveyed elevations, with optional break lines and bounding areas * provides line-of-sight	

Table 8 Functional capabilities for 3D visualization and representation

GRASS 6.3.0	SAGA 2.0.0	ILWIS 3.4
* 3D rendering: - NVIZ: nviz-visualization and animation tool for GRASS data - NVIZ fly through path: creates fly through script to run in NVIZ	* 3D-View: - Setting properties - Rotation: The perspective graphic can be moved and rotated about three axes - Shift: The perspective graphic can be shifted or moved along the three axes - Setting exaggeration - Central projection - Setting perspective distance - Interpolated colours - Anaglyph - Setting eye distance - Interpolated background colour - Interpolated resolution	* Display 3D: edit a 3D georeference to display a map - 3D view with raster draping - Adding vector layers to a 3D view * Apply 3D:apply a 3D georeference on a map * DEM visualization * surface draping of raster images
	- Save as image: This option is used to save a '3D-View' view window as an image - Sequencer: The 'Sequencer' is used to save sets of perspective parameters that can later be used by SAGA to create fly-through image loops	and vector data for 3D visualisation purposes.

Software quality is a level that software has an idealistic combination of attributes (such as reliability, performance and …) [16]. The attributes of significant systems and the best developing methods of them can be considered as:

- Performance
- Reliability
- Usability
- Scalability
- Security.

Often systems fail in providing user requirements (quality loss) and this is when designers have slight attention in providing some requirements, have no attention to the other requirements or are behind schedule in development process.

This is not a new problem and software developers encounter difficulties with it for a long time. As Boehm et al. [17] explain: "finally we concluded that calculation and identification of a general single metric for software quality can be more expensive than its value. Main issue is that most of single quality attributes conflict; often users have problem in identification of their idealistic qualities in these inconsistent conditions".

For non-functional evaluation of FOSS GIS this paper focuses on the performance, scalability and usability tests. Test bed for these tests is a laptop with the following configuration:

- Windows XP Service Pack 2
- Dual core processor
- 1.86 GHz CPU
- 2 GB DDR2 SDRAM
- 160 GB (5400 rpm) HDD
- 1 MB Cache.

6.1 Performance test

As software performance is a widespread concept, therefore for software performance test, different parameters can be evaluated and on that basis one can decide about software performance. In this research two parameters are assessed. The first is "Execution time" and the second "CPU time".

Based on the diagrams in Fig. 1, the following results can be concluded. Execution time of GRASS in processing functions such as DEM, TIN, Contour and Hillshade is very low. Meanwhile, changes in data volume do not have any special influence on the software execution time. But in data import it does not show a good status and as is depicted in Fig. 1 by adding data volumes execution time will grow linearly. Diagrams for

GRASS show that for TIN and DEM functions execution time does not change saliently by adding data volume, but in this case with considering CPU processor time it can be found that any increase in data volume increases CPU processor time.

SAGA has a good status in DEM generation analysis. This software for Hillshade and Data import has a reasonable condition and when data volume increases, execution time of software for these functions increases. But this software does not have an acceptable condition in Contour and TIN functions, because as it can be seen from Fig. 1 by adding to data volume, execution time of software of these functions increases dramatically.

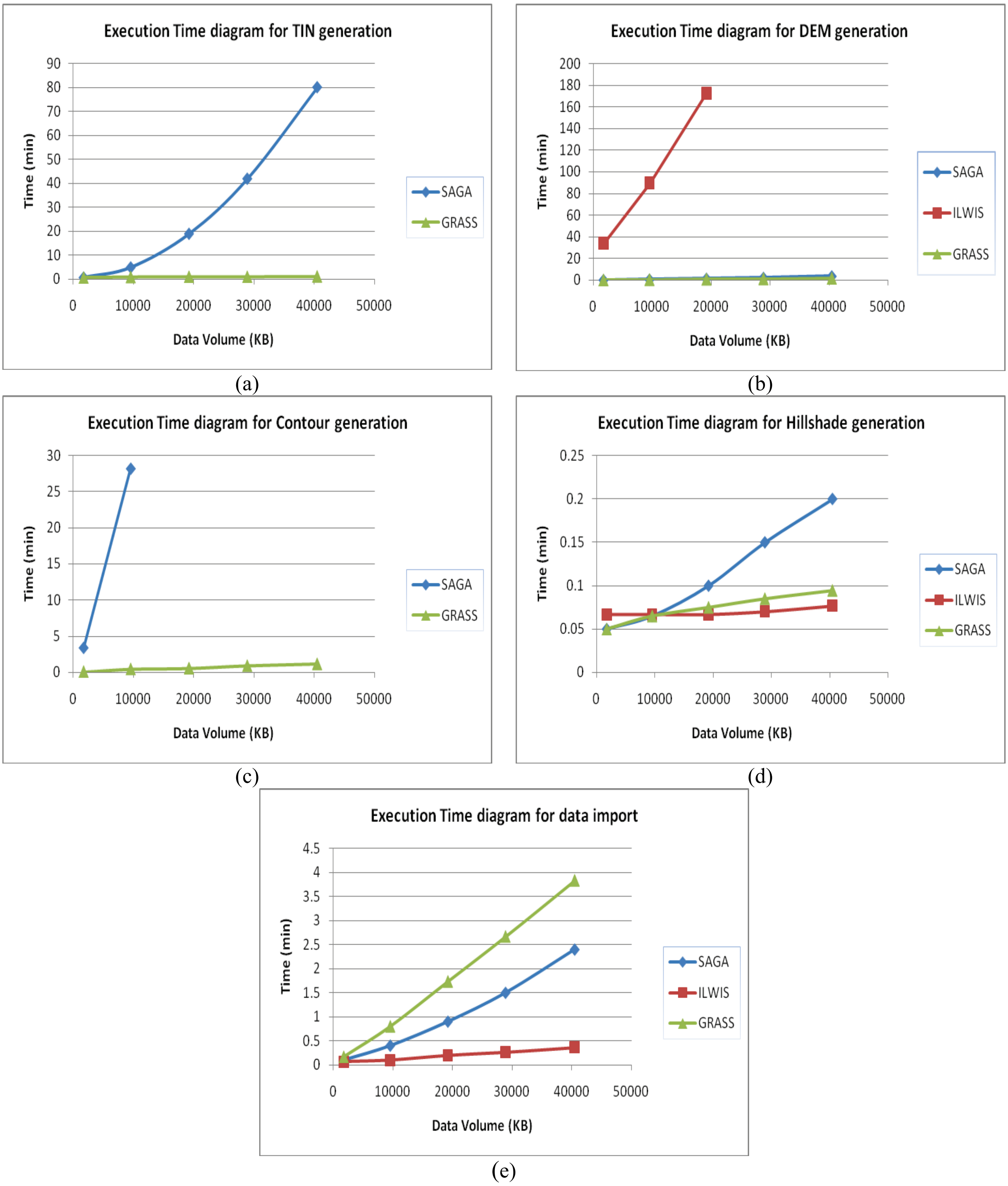


Figure 1 Execution time for different functions of studied software

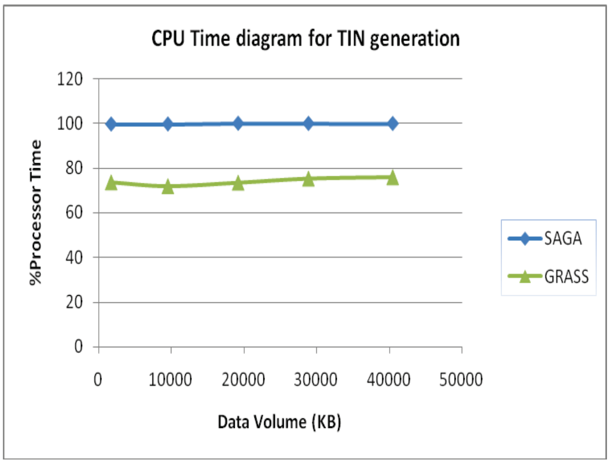
But in ILWIS case it must be mentioned that this software has not provided TIN and Contour functions. Also in DEM generation function it does not have good condition. As it can be seen from the above figure, data volume increase causes a high increase in execution time of software. Remarkable point in this case is that in Hillshade and data import this software has a better status than the other software.

Diagrams in Fig. 2 show the results of subsequent evaluations. Remarkable note with regard to GRASS is that this software in its functions execution always occupies less CPU space than the other software.

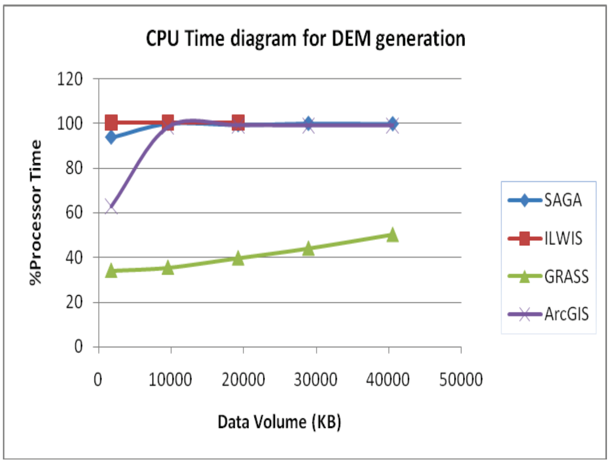
Generally speaking, one can tell that this software from functions execution point of view, always occupies 40 to 80 % of CPU time.

According to the results, SAGA has a constant trend in DEM, TIN, Contour and Data import functions execution which is in all of them this software occupies a high percent of CPU time (between 90 to 100 %). Only in Hillshade function, it has more variability and changes between 20 to 100 % according to the changes in data volume.

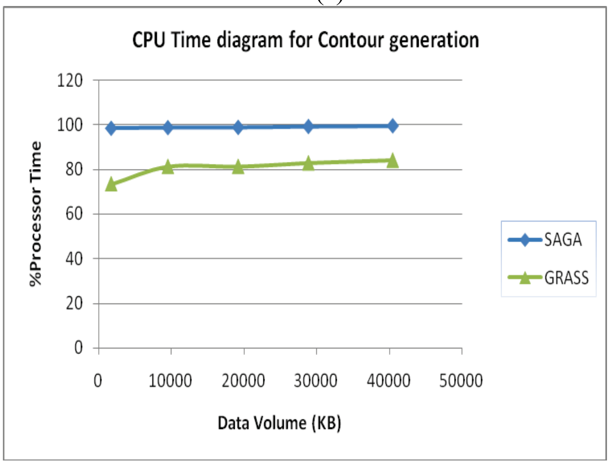
In ILWIS case it has to be mentioned that DEM generation function occupies a high percent of CPU time. But for Hillshade and Data import, results show that with increase in data volume, CPU time occupancy increases too.



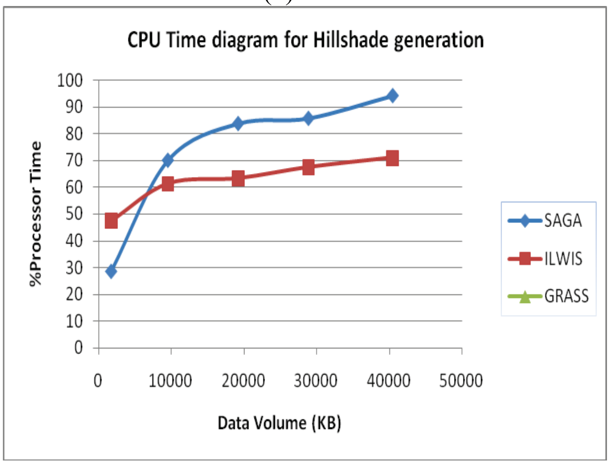
(a)



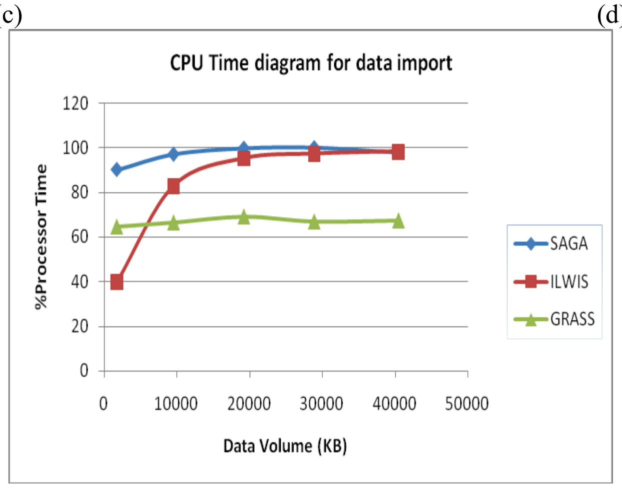
(b)



(c)



(d)



(e)

Figure 2 CPU time for different functions of studied software

The overall result of performance test based on execution time and CPU time depicted that GRASS is well-tuned software and in this category is the best, against SAGA and ILWIS that approximately have similar conditions, SAGA has better performance in execution time test and ILWIS has better performance in CPU time test.

6.2 Scalability test

In telecommunications and software engineering, scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, or to be readily enlarged [18]. Scalability, as a property of systems, is generally difficult to define [19] and in any particular case it is necessary to define the specific requirements for scalability on those dimensions which are deemed important. It is a highly significant issue in electronics systems, database, routers, and networking.

Table 9 Scalability test results for studied software

Scalability	GRASS	ILWIS	SAGA
DEM Generation	Excellent	Poor	Excellent
TIN Generation	Excellent	---	Fair
Contour Generation	Excellent	---	Poor
Hillshade	Excellent	Excellent	Excellent
Data Import	Good	Excellent	Good

In this research, scalability testing of software has been done through different 3D functions of this software. This work is done by different data volume and their results have been presented in Tab. 9.

In the above table, the software analysis results are classified into four qualitative categories. It has to be mentioned that valuation criterion of functions execution is their scalability.

According to test results as it can be seen in the Tab. 9, in scalability testing, GRASS has a better situation than SAGA and ILWIS. Between ILWIS and SAGA it can be concluded that SAGA has a more acceptable condition than ILWIS. Because of Lack of some functions in ILWIS, in some rows of the table there are not any values for ILWIS.

6.3 Usability test

Usability is a field software engineering that is concerned generally with human-computer interaction and specifically with making human-computer interfaces that have high usability or user friendliness. In other words, a user-friendly interface is one that allows users to effectively and efficiently accomplish the tasks for which it was designed and one that users rate positively on opinion or emotional scales.

Usability testing is a technique used to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system [20].

Usability testing measures the usability, or ease of use, of a specific object or set of objects. For this research, a questionnaire is prepared so that its questions are about different usability parameters (listed in table 10). Then 15 students are requested to work with the software under study and fill this questionnaire. Tab. 10 is prepared based on their answers as well as researchers experiences for usability test of the software under study.

Tab. 10 categorizes the results of software analysis results into four qualitative classes. As this is done to determine the usability of the software, therefore, it is necessary to have some effective parameters in usability to do the test based on them. Since there is not such specific criterion for usability, therefore, researchers based on their knowledge, studying and experiences extracted some important usability parameters.

Table 10 Usability test results for studied software

Usability	GRASS	ILWIS	SAGA
Different data formats	Excellent	Good	Excellent
User interface suitability	Fair	Good	Excellent
Consecutive steps to do analysis	Fair	Fair	Good
Default maintenance	Poor	Good	Excellent
Installation easiness	Good	Excellent	Excellent
Icons arrangement	Good	Excellent	Excellent
Software Help	Fair	Excellent	Poor

According to the usability test results shown in Tab. 10, among Free/Open source GIS software in this research SAGA has the best situation. After SAGA, ILWIS and GRASS stand on the second and third ranks, respectively.

7 Discussion

Some results of evaluation in functional capabilities can be summarized as the following:

- In 3D data import, export, GRASS and SAGA can compete with proprietary software.
- In 3D analysis capabilities also GRASS and SAGA are capable to compete with proprietary software whereas in some aspects even GRASS has more functionality.

In 3D representation and visualization SAGA and ILWIS are almost similar. However, in this case GRASS does not have good capabilities.

The results of non-functional evaluation can also be summarized as the following:

- In performance test based on "execution time", results show that GRASS, SAGA and ILWIS have first, second, and third rank, respectively.
- In performance test based on "CPU time", results show that GRASS, ILWIS and SAGA have first, second, and third rank, respectively.
- In scalability test, best scalability is for GRASS. After GRASS, SAGA and ILWIS have the second and third ranks, respectively.
- In usability test, results show that SAGA has the best usability and ILWIS and GRASS stand in the next places in this field.

With all of these, still FOSS GISs are not in their real position. It is mainly due to:

- Lack of a good user interface is an important element that people rarely use this software. Maybe its reason is that proprietary software tries to satisfy users in using their software but FOSS tries to solve problem and user satisfaction comes in the next precedence.
- Often FOSSs executions rely on command line not on a graphic user interface.
- Principally, in FOSS GISs field there isn’t any comprehensive software that can operate in all

aspects such as desktop, and/or web. This is due to two reasons. First in FOSS category the projects usually start with definite goals to scholar these goals. Second, creation of comprehensive software has a high cost and usually FOSS projects are not capable to comply with these expenses.

8 Conclusion

With regard to research motivation, it is seen that growth and development of Desktop FOSS GISs is very considerable. In this part, there are suitable solutions that in plenty of applications, users become needless from proprietary paradigms. To verify this growth and these software conditions we extracted some important assessment criteria, and then based on them were designed some functional and non-functional tests. We also evaluated the proposed tests experimentally. Our experiments showed interesting results. From data analysis view point, GRASS- especially with its graphical user interface and windows based version- is efficient software.

This research specially concentrated on Desktop FOSS GIS software and we assessed different parameters that have impacts on software quality and explicitly defined and evaluated functional and non-functional capabilities that had not been done before. Each of previous works in this field was related to one of application domains and we did our evaluation for 3D analysis capabilities of some selected software.

The results of functional evaluation show that GRASS as a Free/Open source software competes with powerful proprietary software. Moreover, the results of non-functional evaluation show that with regard to different tests that have been done, GRASS, SAGA and ILWIS present acceptable results.

Because subject of Free/open source software in GIS filed is novel there are different research directions that we plan as the following topics for future works:

- A comparing analysis between Free/open source GIS software in the other application domains such as raster analysis, vector analysis, response quality in web including speed, number of clients etc.
- A comparing analysis between free/open source GIS software on different platforms
- Evaluating free/open source GIS software with the other quality parameters such as reliability, extendibility, testability, etc.

9 References

[1] Confino, J. P.; Laplante P. A. An open Source Software Evaluation Model. // International Journal of Strategic Information Technology and Applications. 1, 1(2010), pp. 60-77.

[2] Wheeler, D. A. How to Evaluate Open Source Software Free Software (OSS/FS) Programs. // The Web version (2012), August 18, URL: http://www.dwheeler.com/ oss_fs_eval.html.

[3] Kennedy, M. Evaluating Open Source Software. // The Journal of Software Technology. 14, 1(2011), pp. 1-13.

[4] Mitasova, H.; Neteler, M. GRASS as open source free software GIS: Accomplishments and perspectives. // Transactions in GIS. 8, 2(2004), pp. 145-154.

[5] Badard, T.; Braun, A. OXYGENE: An open framework for the deployment of geographic web services. // In Proceedings of the 21st International Cartographic Conference, Durban, South Africa, CD-ROM. 2003.

[6] Pebesma, E. J. Multivariable geostatistics in S: the gstat package. // Computers and Geosciences. 30, 7(2004), pp. 683-691.

[7] Burghardt, D.; Neun, M.; Weibel, R. Generalization services on the web-classification and an initial prototype implementation. // Cartography and Geographic Information Science. 32, 4(2005), pp. 257-268.

[8] Buliung, R.N.; Remmel, T.K. Open source, spatial analysis, and activity-travel behaviour research: capabilities of the aspace package. // Journal of Geographical Systems. 10, 2(2008), pp. 191-216.

[9] Steiniger, S.; Bocher, E. An Overview on Current Free and Open Source Desktop GIS Developments. // International Journal of Geographical Information Science. 23, 10(2009), pp. 1345-1370.

[10] Ramsey, P. PostGIS Case Studies. Presentation. // Available from: http://www.refractions.net,expertise, whitepapers,postgis-case-studies, (Accessed December 17, 2011).

[11] FSF, The Free Software Definition. // Available from: http:,,www.fsf.org,licensing,essays,free-w.html. (Retrieved October 15, 2011).

[12] Golden, B. Succeeding with Open Source. Addison Wesley, 2005.

[13] Dresen E. The Potential Use of Open Source GI Software in Education Research Semester – Project, FIT, 2006.

[14] Neteler, M.; Mitasova, H. Open Source GIS: A GRASS GIS Approach, 406 illus., Hardcover. p. 80.

[15] Moreno-Sanchez, R.; Anderson, G.; Cruz, J.; Hayden, M. The potential for the use of Open Source Software and Open Specifications in creating Web-based cross-border health spatial information systems. // IJGIS Journal. 21, 10(2007), pp. 1135-1163.

[16] IEEE Standard 1061-1992, Standard for a Software Quality Metrics Methodology. New York, Institute of Electrical and Electronics Engineers, 1992.

[17] Boehm, B. W.; Brown, J. R.; Kaspar, H.; Lipow, M.; MacLeod, G. J.; Merrit, M. J. Characteristics of Software Quality. American Elsevier, Amsterdam, North-Holland Pub. Co., 1978.

[18] Bondi, A. B. Characteristics of scalability and their impact on performance. // Proceedings of the 2nd international workshop on Software and performance, Ottawa, 2000, pp. 195-203.

[19] Duboc, L. et al. Doctoral symposium: presentations: A framework for modelling and analysis of software systems scalability. In Proceeding of the 28th international conference on Software engineering ICSE, 2006, pp. 949-952.

[20] Nielsen, J. Usability Engineering. Academic Press Inc, 1994, p. 165.

Authors' addresses

Mohammad Akbari
Deptartment of Geomatics Engineering, University of Tehran
Tehran, 14665-331, Iran
E-mail: moakbari@ut.ac.ir, mohammad.akbari@geo.uzh.ch
Tel/fax: +98 21 8833 4341, Cell: +98 915 361 6696

Mohammad Ali Rajabi
Assistant Professor of Geospatial Information Systems
Department of Geomatics Engineering, University of Tehran
Tehran, 14665-331, Iran
Tel/fax: +98 21 8833 4341, Cell: +98 915 361 6696