

# About the life cycle of information systems based on open Web-technologies

Sergey Kratov and Olga D Sokolova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,  
Novosibirsk, Russia

[kratov@sscc.ru](mailto:kratov@sscc.ru)

**Abstract.** The article describes the full life cycle of an information system based on open Web-technologies from the point of view of developers and system administrators. The article starts with the development of technical specifications for an information system creation and ends with the system's launching into operation and its subsequent uninterrupted operation. As the example, the creation process of the information system for the Foundation of Algorithms and Programs of the Siberian Branch of the Russian Academy of Sciences is described.

## 1. Introduction

Information technologies today are the base and basis for a wide range of research in all scientific areas. Software developed in course of scientific research has both a number of general basic laws for implementation of information processes and specific differences due to difference in information environments in which this software operates. In modern conditions, software development cycle is significantly reduced. But software must be able to work stably on a variety of hardware architectures and software environments. And in the conditions of software code openness, detected errors can be used to deliberately disrupt its operation. Accordingly, the tasks of creating public catalogs and standardized software repositories are becoming critically important, contributing to the expansion of the software users' circle and accelerating the delivery time of its new versions to users.

Over the past two decades, the world has accumulated extensive experience in the field of creating software repositories and remote team software development [1-3]. A significant part of software is then distributed as free and open source. In research centers and universities, in course of research, a large number of science-intensive and high-tech software of a wide scientific profile is also being developed [4-5]. Unfortunately, most of the developed software of this class is then used only within research centers, workplaces of its authors, or even within individual laboratories. Refinement of high science-intensive software to the level of completed software products with their subsequent inclusion in global software repositories would significantly increase the number of its potential users. It would also increase the return on the financial and time expenditures of researchers spent on its development, provide additional opportunities for testing and upgrading such software by third-party developers and users [6-7].

To solve this problem, the Siberian Branch of the Russian Academy of Sciences decided to create the Foundation of algorithms and programs intended for the accumulation, systematization, storage and use of science-intensive software products developed at the institutes of the SB RAS. The prerequisites for the creation of such the catalog of programs and databases was the almost complete

lack of information about the new software developments of scientists. When creating the Foundation, the following tasks of its functioning were determined:

- Expertise of software products developed in organizations of the SB RAS and submitted to the FAP SB RAS.
- Registration and cataloging of such algorithms, software and information resources. Providing access to them.
- Organizational and informational support for the distribution of software products registered with the Foundation.

To solve these problems, both the already known methods of software classification and the organization of its ordered storage and access to it were used, and new approaches were developed. Methods for creating open software systems and platforms, developed and published by the authors of the article earlier, were improved and applied.

Despite the fact that the Foundation of algorithms and programs was created by the Siberian Branch of the Russian Academy of Sciences, developers from various research centers and universities in Russia register their programs and databases in it. During the creation of the Foundation, the analysis of the existing global repositories of software packages was carried out. The classifier of high-tech software was developed and on its basis the repository of high-tech and science-intensive software was organized in the format of open operating system packages (deb, rpm) based on the Siberian Supercomputer Center. The created repository made it possible to reuse the previously developed science-intensive software in the process of conducting scientific research. Also, the methodology was created for finalizing science-intensive software to the level of ready-made software products with their subsequent inclusion in global repositories. This, in turn, made it possible to disseminate the received practices among scientific groups of different profiles, one of the results of scientific activity of which is software.

The development and operation of the information system of the Foundation of Algorithms and Programs of the SB RAS is described in this article as a practical example of the life cycle of information systems based on Web technologies. The article is divided into several sections corresponding to the main stages of the life cycle. The second section describes the technological development stack (programming language, DBMS), provides the rationale for the choice of development tools (open and free framework). In the third section, the stages of designing an information system and development itself are given, an estimate is given of the timing of the implementation of these stages, depending on external conditions. The fourth section describes the requirements for the server hardware and software environment on which developed information system will be deployed. The fifth section is devoted to the stage of initial testing of information system, elimination of identified errors, and writing user documentation. The sixth and seventh sections are written from the point of view of the administrators who are responsible for the trouble proof operation of system. These two sections are devoted to the longest stage of the life cycle - the direct operation of system. Also, these sections describe the process of ensuring the information security of system. In the eighth section, optional life cycle stages are given. The implementation of this stages is not mandatory, but can accelerate the achievement of the goals of creating information system. The conclusions of the article provide statistical data on the operation of information system of the Foundation of Algorithms and Programs of the SB RAS, obtained during its operation.

## **2. Information system technology stack**

The information system of the Foundation of Algorithms and Programs of the SB RAS was developed based on Drupal content management framework (CMF). Content management frameworks provide usable interfaces for administration and management of the resulting information system's content. And thereby frameworks are often used to create information systems accessible via Web [8-10]. A number of CMF and CMS are available for developers, both free and commercial. Among the most frequently used ones, one can single out such as WordPress, Drupal, Joomla!, 1C-Bitrix.

Based on the set of parameters (kernel functionality, expansion options using external modules, the ability to write custom modules, type of license), the CMF Drupal was chosen to create the information system of the FAP SB RAS [11-13]. Drupal is the free CMF (Content Management Framework) with good scalability and flexibility. For this CMF, third-party developers have written a large number of plugins that extend its standard functionality [14]. The development of the FAP SB RAS information system based on CMS Drupal made it possible to combine all the Foundation's information resources on the basis of the single platform, simplified content placement and the information system administration.

The technological basis of the framework is:

- The development language - PHP (version from 5.x to later).
- Relational DBMS (MySQL, PostgreSQL, etc.).
- Web server (Apache, nginx, IIS, etc.).
- An operating system (OS) of the Linux or Windows families (and others on which the PHP interpreter can run).

The use of such technological platforms [15, 16] made it possible to significantly reduce the time and financial costs of creating the FAP SB RAS information system [17-19]. It also eliminated the need to write and test the basic functionality of the information system.

The 7th and 8th versions of the CMF are relevant nowadays and, therefore, can be recommended for the development. Both versions have their advantages and disadvantages:

- The use of the 7th version is advisable, because it has been the main version of the CMF for several years. During this period, third-party developers have written thousands of modules that implement certain functions of information systems that have already been tested in work, tested as compatible with other software, and for the presence of errors and vulnerabilities in them. At the same time, the 7th version today is already outdated from the developers of the CMF core point of view. For most of modules compatible with this version, only updates related to the correction of found security problems are issued. Functionality of modules is rarely expanded.
- Using the 8th version of the CMF would be result in a more technologically advanced version of the system, since the 8th version is based on modern Symfony2 and Twig frameworks and template engines. But at the same time, a significant part of existing modules was not imported to the 8th version of the kernel and it would be necessary to write its functionality from scratch, which would require additional time and financial costs.

For the combination of the reasons listed above, the 7th version of CMS Drupal was chosen as the kernel for the implementation of the information system.

As one of the options for creating the FAP SB RAS information system, the use of experience and developments of ready-made assemblies based on CMF Drupal was considered, designed to solve similar problems: creating university sites, research laboratories and corporate sites (OpenScholar, Research Lab, Open Atrium, Corporate site). Assemblies consist with the core of CMS Drupal with a predefined set of additional modules (as well as possibly specialized modules developed by the assemblies' authors) designed to create information systems that will solve the above problems. Even if none of the assemblies fully satisfies the functionality required of the resulting information system, their solutions and developments may be useful in the process of developing your own solution.

### **3. Stages of information system's design and development**

The following main stages can be distinguished in the life cycle of an information system:

Drawing up of technical specifications for design / development, in particular:

- Coordination of the system functionality with the all interested parties.
- Development of the technical specifications in accordance with the agreed functionality [20].
- Preparation of the design prototypes for transfer to the designer.

- Discussion of the prototypes and terms of design manufacturing with the designer and officials of the customer [21].
- Allocation of the tasks among the development team, coordination of implementation teams [22-24] and order of interaction between the participants [25, 26].

The duration of the stage of drawing up technical task implementation is from 2 weeks. It largely depends on competence of responsible employee on the part of customer and level of his authority for making decisions about functionality of information system being developed. In the case of the Foundation system, it took 2 weeks to draw up the technical task initial terms for the development. In the process of implementation and further operation of information system, amendments were made to the technical task terms to comply with changing external conditions. Such as changes in the requirements for functionality and the software environment in which the information system operates.

Stage participants are:

- Developers from the ICM&MG SB RAS (the head organization of the FAP SB RAS).
- Representatives of the SB RAS (the customer of the system).
- The deputy director for IT of the ICM&MG SB RAS (general coordination).

The information system software development:

- Allocation of servers or virtual machines for the system development, installation of the OS on that servers.
- Granting the developers access to the servers.
- Development of the information system basic functionality.
- Development of the interface forms and the system design as a whole.
- Development of connectors to the free software repositories (if necessary) [27].
- Development of connectors to the free software demonstration platform (if necessary) [28].
- Development of the user's and system administrator's documentation.

The duration of this stage implementation depends on the complexity of the information system being created. In the case of the Foundation's system, the development of the first working version put into test operation took one and a half months. Stage participants are the administrators and developers on the contractor's side. General coordination is on the deputy director for IT on the contractor's side.

#### **4. The hardware and software environment requirements of the information system**

Based on the experience of a similar class information systems previous developments, the following requirements were made to the server hardware for the operation of the information system:

Hardware or dedicated server:

- CPU: at least 1 dedicated core of the server's processor (preferably a separate processor as a whole).
- RAM: at least 2 GB for the OS as a whole.
- Disk space: at least 10 GB for the OS as a whole. In the case of intensive system operation, the size of the disk space will subsequently need to be increased.
- Network port: 1Gb/s.
- Dedicated IP address.
- External access via HTTPS, FTP and SSH.
- Ability to send and receive e-mails.

Requirements for the software environment are:

Basic environment:

- OS of Linux family (for example, Ubuntu Server 16.04).
- RDBMS (MySQL5.X and PostgreSQL9.X latest stable versions).
- PHP7.X latest stable version (the minimum set of additional modules are libapache2-mod-php7.0, php7.0-curl, php7.0-dom, php7.0-gd, php7.0-mbstring, php7.0-pdo-pgsql).
- Web server (Apache2.X latest stable version).

Additional packages:

- Web interfaces for DBMS (phpmyadmin, phppgadmin).
- Mail server (postfix).
- SSH server (openssh-server).
- Package manager for PHP (composer).
- Version control system (git) [29-31].
- Console interface for Drupal (drush).

Ports:

- Open ports (FTP, HTTP, HTTPS, SMTP, SSH).

The developers and the administrator need the full administrative (root/sudoers) access to the server's software environment or at least:

- the rights to create / edit directories and configuration files of Apache, PHP, MySQL, PostgreSQL,
- the owner's rights to the directories with web server public files,
- the rights to install additional packages (for example, PHP modules),
- the rights in the RDBMS to create databases, users and grant access to them,
- the rights to manage (restart) web server and RDBMS services.

During the work of developers of a specific information system, additional libraries, modules, tools that are not listed in the requirements above may be needed and used.

## 5. Testing of the developed information system

There is the transition to the stage of testing the developed system [32-34] upon completion of the first stages - design and development:

- Organizing test training sessions with users on the system's beta version (without design).
- Testing the system's functionality by the potential users.
- Getting the first versions of the design. Selection and approval of the version, creation of the system's design.
- Collecting of comments on the functionality, system interface and documentation.

The duration of the stage depends on the complexity of information system and competencies of both testers from development team and the responsible employees of customer who accept developed system. Stage participants are:

- The potential users (high-tech software developers).
- The representatives of the SB RAS (the system's customer).

During testing, the information system's version is deployed on a separate server, on which users begin to work. A number of test user accounts with roles (a list of permissions) is created similar to real ones to evaluate the system's functionality and interface for different user roles. These test users are blocked most of the time on the main working version of the system to prevent accidental data corruption/loss. There are used only for a short-term assessment of the changes made.

The system's test version from time to time may not fully coincide with the main working version [35, 36]. If it is necessary to obtain the exact copy, a request should be sent to the technical administrators of the system.

All errors detected during the testing phase are fixed, the agreed additions/changes to the system's functionality are also made, and the corrections are made to its documentation [37, 38]:

- Making corrections agreed with the responsible persons in the system's code and interfaces.
- Commissioning the system's test version.
- Editing the documentation in accordance with the changes.
- Commissioning of the system's final version.

The implementation period is two weeks. During the test operation period (week), the technical and consulting support is provided to users by the contractor's technical staff.

## **6. The system support after putting it into operation**

At the end of the test period, the system is fully commissioned. The system is administered to ensure its smooth and safe functioning as well as the software and hardware environment in which it operates [39, 40]:

- the Web server software support in operational state,
- the Web server and the system administration,
- installation of the OS and other software necessary for the system to work on the Web server,
- the software configuration on the Web server. Configure of the web server, remote administration services, firewalls. Secure the Web server and the system (protection against unauthorized access to information, viewing or changing system files and data) [41-43]. Identify software errors and take fixing actions [44].
- registration of the system's users, assignment of logins and passwords, setting of access rights and control of resources' use (as agreed with the system's editor),
- ensuring timely copying, archiving and backing up the system's data, taking actions to restore the system and the Web server in case of software failures [45, 46],
- traffic analysis, reporting on traffic to the Web server,
- preparation of proposals for the development and modernization of the system's software and hardware complex,
- updating/developing, testing and debugging the software of the system's server part. It provides the ability to perform the system's functions in accordance with the tasks,
- support of the developed software and other software tools of the information system,
- modernization of the system interface (with the involvement of a third-party web designer, if necessary),
- development of the measures for information security of the system and ensuring their implementation (as agreed with the deputy director for IT),
- organization of the system's promotion on the Internet (within the budget allocated for these purposes) [47].

The implementation period is all the time of the system's active operation. Stage participants are network/system administrators of the contractor. If necessary, third-party developers and front-end developers, designers are involved.

## **7. Regular monitoring of the state of the system and its modernization**

After the information system is launched into operation, the main and often the longest stage of its life cycle begins. Successful completion of this stage requires support for system users and involvement of qualified system administrators [48]. For example, for the FAP SB RAS information system, the following set of tasks is performed on the regular basis:

- Configuring software on the system's hosting:
  - Configuring Web server parameters (switching to the latest PHP version compatible with the CMS core used; installing the SSL certificate and forcing the transfer to work using the secure HTTPS protocol),
  - Configuring remote access and administration services (regularly changing passwords of the hosting panel, FTP user, DB owner)) in order to ensure the security of the Web server and the information system itself (protection against unauthorized access to the information, viewing or changing the system files and data) [49-51].
- Conducting the audit of the hosting file system with the removal of files that are not related to the information system's operation, if they are detected (virus or system infected files). Replacing system directories and files with their "clean" copies (from the developers' repositories, CMS modules and third-party libraries) and setting them to the correct access rights.

- Implementation of the full analysis of the information system's logs for the previous period from the time of the previous check:
  - Attempts to access restricted content.
  - Attempts to access non-existent system pages.
  - Work with content (the creation of new materials and editing existing).
  - Messages from the PHP interpreter about warnings and problems that arise during the information system's operation.
  - List of actions that affected system settings.
  - List of actions on users (registration of new ones and change of rights and authorities of the existing ones).
- Audit the information system's list of users with the removal of unauthorized users and user roles and the rollback of their actions. Change passwords of the users with advanced privileges (the system editor, the CMS administrator). Performing verification of access rights for various user roles and monitoring user activity for the previous 2-week period.
- Update to the latest versions of the CMS core, all additional modules and libraries used in the work of the information system (the SPAM protection, the visual editor, the image handlers, the interactive menus). Conducting compatibility checks for new versions of the system's hosting software.
- Issuing recommendations to the information system's users with administrative rights on the implementation of the information system's state regular monitoring:
  - Tracking the relevance of the kernel versions and additional CMS modules, updating them timely in case security patches are released from their developers.
  - Tracking the relevance of third-party libraries' versions in the hosting's file manager, timely updating them in case of security patches' release from their developers.
  - View the list of users for new ones and changing roles for previously registered users.
  - View the list of operations on the content - the appearance of new content and editing the old.
  - View shared logs.
  - Organizing the regular backup of the information system's content in order to ensure the recovery of the information system during software failures and/or infection/hacking.

Fulfillment of these conditions makes it possible to ensure uninterrupted and safe functioning of the system throughout the entire period of its operation up to the moment of termination of its operation.

## **8. Development of system's functionality**

Further stages of the system's life cycle can be, for example, such as:

- Improvement of the system's functionality in accordance with the recommendations collected in the starting working period.
- Implementation of secondary functions that enhance usability. For example, such as:
  - Development of the system's versions for various mobile devices (separation depending on screen resolution and computing power) and OS running on them (iOS, Android, etc.) [52-54].
  - Organization of permanent and public export of announcements of the system's content divided into thematic RSS (Really Simple Syndication) feeds, as well as reverse news' integration from sites of similar subjects [55-57].
  - Organization of mailing lists for formed and constantly updated user groups.
  - Collection of statistics about system's visits in general, determining the most popular content, statistics on file downloads, visitors' sources (search engines, social networks, other sites, etc.) and other parameters. The regular analysis of views' statistics of individual content and the system as a whole.

The stage implementation period depends on the development options.

## 9. Conclusion

The practical result of the FAP SB RAS information system operation is the expertise with the subsequent registration and cataloging in the Foundation of more than 260 science-intensive software and more than 30 databases. During the expertise process, an average of 10% of applications were rejected. The result of the registration of a software product or database in the Foundation is an organizational and informational support for their distribution. The certificate of registration in the Foundation does not give any legal status, so the formalization and registration of approximately 25% of applications were not completed by the authors. Twenty registered programs and databases were transferred to the archive of the Foundation at the request of the authors / applicants.

Most of the software and database admitted to the Foundation were registered in the first years of the Foundation's work. At present, the dynamics of registration of software and databases in the Foundation, unfortunately, is negative. The main reason for this dynamic is the lack of organizational and financial support for the activities of the Foundation by higher authorities. Nevertheless, at the moment in the information system of the Foundation, about 60 applications are in the process of editing by the authors, 16 applications have been signed by the authors and sent for examination. The average traffic to the pages of the Foundation's information system today is about 10,000 per week.

## Acknowledgments

This work was supported by the budget project of the ICM&MG SB RAS no. 0251-2021-0005.

## References

- [1] Wu Y H, Wang S W, Bezemer C P and Inoue K 2019 How do developers utilize source code from Stack Overflow? *Empirical Software Engineering* **24(2)** 637–73
- [2] Gao Y Q and Madey G 2007 Network analysis of the SourceForge.net community *Open Source Development, Adoption and Innovation* **234** 187
- [3] Voth D 2000 CodePlex: Microsoft's latest entry into code sharing *IEEE Software* **23(5)** 96–8
- [4] Friedrichs M, Shoshi A, Chmura P J, Ison J, Schwammle V, Schreiber F, Hofstadt R and Sommer B 2019 JIB.tools 2.0 - A Bioinformatics Registry for Journal Published Tools with Interoperability to bio.tools *J. of Integrative Bioinformatics* **16(4)** 20190059
- [5] Li Q L, Zhou Y C, Jiao Y M, Zhang Z, Bai L, Tong L, Yang X, Sommer B, Hofstadt R and Chen M 2016 DaTo: an atlas of biological databases and tools *J. of Integrative Bioinformatics* **13(4)** 297
- [6] Thelwall M and Kousha K 2016 Academic software downloads from Google Code: useful usage indicators? *Inf. Research-An Int. Electronic J.* **21(1)** 709
- [7] Cheng X F, Zhang Z M, Yang Y and Yan Z H 2020 Open collaboration between universities and enterprises: a case study on GitHub *Internet Research* **30(4)** 1251–79
- [8] Taylor L, Rusack E and Zemleris V 2012 A new information architecture, website and services for the CMS experiment *J. of Physics Conf. Series - Int. Conf. On Computing In High Energy And Nuclear Phys. (CHEP2012)* vol 396 (New York: New York Univ) p 062021
- [9] Kratov S 2018 On the approaches to the development of a scientific organization website *Proc. of the 2018 3rd Russian-Pacific Conf. on Computer Technology and Applications (RPC)* (New York: IEEE) p 8482177
- [10] Atayero A A, Chijioke-keme and Ogunjobi B 2014 FOSS implementation of an educational virtual office suite *EDULEARN14: 6th Int. Conf. on Education and New Learning Technologies* ed L G Chova and A L Martinez et al (Barcelona) pp 5018–27
- [11] Caron B, Toole D, Wicks P and Miller S 2011 Digital Ocean: building a platform for scientific collaboration and social and media sharing on the Drupal content management system *Earth Science Informatics* **4** 191–6
- [12] Kratov S 2017 The information system for the scientific news site support *2017 Int. Multi-Conf. on Engineering, Computer and Information Sciences (SIBIRCON)* (New York: IEEE) p 8109848



- [13] Drouillon P J and Dutre P 2012 How to (re-) build a frequent asked questions site with Drupal, an open-source CMS *5th Int. Conf. of Education, Research and Innovation (ICERI 2012)* ed L G Chova and A L Martinez et al (Madrid) pp 908–12
- [14] Tramullas J 2013 Content management with Drupal: a review of modules specific to libraries, archives and museums *Profesional De La Informacion* **22** 425–31
- [15] Zybarev Y and Kratov S 2019 The project office for bioresource collections management *SIBIRCON 2019 - Int. Multi-Conf. on Engineering, Computer and Information Sciences, Proceedings* (New York: IEEE) pp 657–60
- [16] Kratov S and Zybarev Y 2020 To the question of the digital platform "Bioinformatics" creating and its system-forming solutions *J. Of Integrative Bioinformatics* **17(4 SI)** 20200029
- [17] Sokolova O D and Kratov S V 2016 Information systems for popularization of scientific and knowledge-based software *2016 13th Int. Scientific-Technical Conf. on Actual Problems of Electronic Instrument Engineering (APEIE)* vol 2 (New York: IEEE) pp 519–22
- [18] Sokolova O and Kratov S 2018 Foundations of algorithms and programs: history and prospects *2017 Fourth Int. Conf. on Computer Technology in Russia and in the Former Soviet Union (SoRuCom)* ed I Krayneva and A Tomilin (New York: IEEE) pp 120–2
- [19] Zybarev Y and Kratov S 2012 The information support system of corporate software resources development *Proc. - 2012 7th Int. Forum on Strategic Technology (IFOST 2012)* p 6357649
- [20] Ali K and Xia X 2018 Revisiting and rethinking on the technical aspects of software requirements *Proc. of the Int. Conf. on Information Technology And Electrical Engineering 2018 (ICITEE '18)* ed S Patnaik (New York: Assoc Computing Machinery) p 3306274
- [21] Gomero-Fanny V, Bengy A R and Andrade-Arenas L 2021 Prototype of web system for organizations dedicated to e-commerce under the SCRUM methodology *Int. J. of Advanced Computer Science and Applications* **12(1)** 437–4
- [22] Ortu M, Destefanis G, Kassab M and Marchesi M 2015 Measuring and understanding the effectiveness of JIRA developers communities *Sixth Int. Workshop on Emerging Trends in Software Metrics (WETSOM)* (New York: IEEE) pp 3–10
- [23] Kratov S and Bukhtiarov I 2013 The technological platform for software development in the SB RAS *8th Int. Forum on Strategic Technology 2013, IFOST 2013 – Proc.* vol 2 pp 353–5
- [24] Kanai J and Anderson M 2013 Redmine as a web-based collaboration tool in engineering design courses *2013 ASEE Annual Conference* (Washington: Amer Soc Engineering Education)
- [25] Santos Júnior P S, Barcellos M P, Falbo R D A and Almeida J P A 2021 From a SCRUM reference ontology to the integration of applications for data-driven software development *Inf. and Software Tech.* **136** 106570
- [26] Senapathi M and Drury-Grogan M L 2021 Systems thinking approach to implementing Kanban: A case study *J. of Software: Evolution and Process* **33(4)** e2322
- [27] Sokolova O and Kratov S 2020 Platforms for joint development and hosting of software and the example of their implementation in the FAP SB RAS *Proc. - 2020 Int. Conf. Engineering Technologies and Computer Science, EnT 2020* ed S Prokhorov (New York: IEEE) pp 20–3
- [28] Kratov S V 2011 The free software demonstration platform *Proc. of the 6th Int. Forum on Strategic Technology, IFOST 2011* vol 2 pp 1331–3
- [29] Begel A, Bosch J, Storey M-A, Doll B, Laundry D, Fullerton D and Hughes R 2013 Social networking meets software development: perspectives from GitHub, MSDN, Stack Exchange, and TopCoder *IEEE Software* **30(1)** 52–66
- [30] Sims J and Woodard C J 2020 Community interactions at crowd scale: hybrid crowds on the GitHub platform *Innovation-Organization & Management* **22(2)** 105–27
- [31] Tushev M, Williams G and Mahmoud A 2019 Using GitHub in large software engineering classes. An exploratory case study *Computer Science Education* **30(2)** 155–86
- [32] Sanchez A B, Segura S, Parejo J A and Ruiz-Cortés A 2017 Variability testing in the wild: the Drupal case study *Software And Systems Modeling* **16** 173–94
- [33] Chimos K and Loumpardas P 2014 Identifying security issues in a higher education's institute

- CMS lab site *Edulearn14: 6th Int. Conf. On Education And New Learning Technologies* (Valencia: Iated-Int Assoc Technology Education A& Development) pp 3349–54
- [34] Javed A and Schwenk J 2014 Towards elimination of cross-site scripting on mobile versions of web applications *Information Security Applications, WISA 2013* vol 8267, ed Y Kim and H Lee et al (Cham: Springer Int Publishing) pp 103–23
- [35] Machols J 2004 The Subversion version-control program *Dr Dobbs J.* **29(8)** 64
- [36] Prahov R, Schmidt H and Graupner A 2012 Subversion(r): an empirical performance case study from a collaborative perspective on integrated circuits and software development *Proc. of 2013 IEEE 4th Int. Conf. on Software Engineering and Service Science (ICSESS)* ed M S P Babu and L Wenzheng (New York: IEEE) pp 35–42
- [37] AlOmar E A, Peruma A, Mkaouer M W, Newman C, Ouni A and Kessentini M 2021 How we refactor and how we document it? On the use of supervised machine learning algorithms to classify refactoring documentation *Expert Systems With Applications* **167** 114176
- [38] Marmolejos L, AlOmar E A, Mkaouer M W, Newman C and Ouni A 2021 On the use of textual feature extraction techniques to support the automated detection of refactoring documentation *Innovations In Systems And Software Engineering*
- [39] Kratov S 2018 On providing the fault-tolerant operation of information systems based on open content management systems *Xi Int. Scientific and Technical Conf. - Applied Mechanics And Dynamics Systems* vol 944(1) (Bristol: IOP Publishing Ltd) p 012067
- [40] Patel S K, Rathod V R and Prajapati J B 2013 Comparative analysis of web security in open source content management system *2013 Int. Conf. on Intelligent Systems and Signal Processing (ISSP)* (New York: IEEE) pp 344–49
- [41] Martin-Galan B, Hernandez-Perez T and Rodriguez-Mateos D and Pena-Gil D 2009 The use of robots.txt and sitemaps in the Spanish public administration *Profesional De La Informacion* **18(6)** 625–30
- [42] Kolay S, D'Alberto P, Dasdan A and Bhattacharjee A 2008 A larger scale study of robots.txt. *Proc. of the 17th Int. Conf. on World Wide Web 2008, WWW'08* pp 1171–2
- [43] Tong W and Xie X 2009 A research on a defending policy against the webcrawler's attack *Proc. of the 3rd Int. Conf. on Anti-Counterfeiting, Security, and Identification in Communication* (New York: IEEE) pp 363-6
- [44] Serrano N and Ciordia I 2005 Bugzilla, ITracker, and other bug trackers *IEEE Software* **22(2)** 11–3
- [45] Kim B-H 2021 Design of reliable disaster recovery system through integrated server redundancy *Turkish J. of Computer and Math. Education* **12(6)** 674–9
- [46] Banas V, Babarada F and Ravariu C 2020 Windows server backup and restore for Moodle e-learning platform *Proc. of the 2020 12th Int. Conf. on Electronics, Computers and Artificial Intelligence (ECAI-2020)* (New York: IEEE)
- [47] Kratov S 2019 On improving the indexing of information systems based on web technologies *XII Int. Scientific and Technical Conf. Applied Mechanics and Systems Dynamics* vol 1210 (Bristol: IOP Publishing Ltd) p 012072
- [48] Eshkevari L, Dos Santos F, Cordy J R and Antoniol G. 2015 Are PHP applications ready for hack? *2015 22nd Int. Conf. on Software Analysis, Evolution, and Reengineering (SANER)* ed Y G Gueheneuc and B Adams et al (New York: IEEE) pp 63–72
- [49] Kratov S 2019 About leaks of confidential data in the process of indexing sites by search crawlers *Lecture Notes in Computer Science* vol 11964, ed N Bjorner and I Virbitskaite et al (Cham: Springer International Publishing) pp 199–204
- [50] Zhang Y, Lo D and Xia X, Xu BW, Sun JL and Li SP 2015 Combining software metrics and text features for vulnerable file prediction *2015 20th Int. Conf. on Engineering of Complex Computer Systems (ICECCS)* (New York: IEEE) pp 40–9
- [51] Mainka C, Mladenov V and Schwenk J 2016 Do not trust me: using malicious IdPs for analyzing and attacking single sign-on *Proc. - 2016 IEEE European Symposium on Security*

- and Privacy, EURO S and P 2016* (New York: IEEE) pp 321–36
- [52] Skerrett D 2019 Ten years of the App Store: what can we learn from our mobile past? *Econtent* **42(2)** 12–3
- [53] Businge J, Openja M, Kavaler D, Bainomugisha E, Khomh F and Filkov V 2019 Studying Android app popularity by cross-linking GitHub and Google Play store *2019 IEEE 26th Int. Conf. on Software Analysis, Evolution and Reengineering (SANER)* ed X Wang D Lo et al (New York: IEEE) pp 287–97
- [54] Fan Q, Wang T, Yang C, Yin G, Yu Y and Wang H M 2019 Why do they ask? An exploratory study of crowd discussions about Android application programming interface in Stack Overflow *J. of Central South University* **26(9)** 2432–46
- [55] Okolo S E 2019 Web 2.0 for reference services: an overview *Library Philosophy and Practice* **2019** 2390
- [56] Elyusufi Y and En-Naimi E M 2017 User profiling and RSS feeds recommendation based on the ontological approach *ICCWCS'17: Proc. of the 2nd Int. Conf. on Computing and Wireless Communication Systems* (New York: Assoc Computing Machinery) p a10
- [57] Singh G and Sahu S 2015 Review on "Really simple syndication (RSS) technology tools" *2015 IEEE Int. Conf. on Computational Intelligence and Communication Technology CICT 2015* (New York: IEEE) pp 757–61