Asociaţia Grupul pentru Reformă şi Alternativă Universitară (GRAUR)
Cluj-Napoca
Indexul Operelor Plagiate în România
www.plagiate.ro

Decizie de indexare a faptei de plagiat la poziţia
00366 / 22.03.2017
şi pentru admitere la publicare în volum tipărit

care se bazează pe:

A. Nota de constatare şi confirmare a indiciilor de plagiat prin fişa suspiciunii inclusă în decizie.

Fişa suspiciunii de plagiat / Sheet of plagiarism's suspicion	
Opera suspicionată (OS)	**Opera autentică (OA)**
Suspicious work	**Authentic work**
OS	VINTILĂ, Cristian, and SCHNAKOVSZKY, Carol. The using of the POKA-YOKE techniques in order to avoid the translation programs and meniu from other languages, *Modelling and Optimization in the Machines Building Field (MOCM)*. 8. 2002. p. 144-149.
OA	Harry ROBINSON. Using Poka-Yoke Techniques for Early Defect Detection. Sixth International Conference on Software Testing Analysis and Review (STAR'97). San Jose. SUA. 7 Mai 1997. www.faculty.washington.edu/apurva/502/Readings/Lean/pokasoft.pdf.
Incidenţa minimă a suspiciunii / Minimum incidence of suspicion	
p.144:06 – p.149:19	p.1:06 – p.1:16; p.1:19 – p.1:24; p.2:24 – p.2:32; p.2:35 – p.3:10; p.4.01 – p.5:07; p.5:10 – p.6:21; p.6:23 – p.6:25; p.7:03 – p.10:08; p.10:23 – p.11:02
Fişa întocmită pentru includerea suspiciunii în Indexul Operelor Plagiate în România de la Sheet drawn up for including the suspicion in the Index of Plagiarized Works in Romania at www.plagiate.ro	

Notă: Prin „p.72:00" se înţelege paragraful care se termină la finele pag.72. Notaţia „p.00:00" semnifică până la ultima pagină a capitolului curent, în întregime de la punctul iniţial al preluării.

Note: By „p.72:00" one understands the text ending with the end of the page 72. By „p.00:00" one understands the taking over from the initial point till the last page of the current chapter, entirely.

B. Fişa de argumentare a calificării de plagiat alăturată, fişă care la rândul său este parte a deciziei.

Echipa Indexului Operelor Plagiate în România

Asociaţia Grupul pentru Reformă şi Alternativă Universitară (GRAUR)
Cluj-Napoca
Indexul Operelor Plagiate în România
www.plagiate.ro

Fişa de argumentare a calificării

Nr. crt.	Descrierea situaţiei care este încadrată drept plagiat	Se confirmă
1.	Preluarea identică a unor pasaje (piese de creaţie de tip text) dintr-o operă autentică publicată, fără precizarea întinderii şi menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	✓
2.	Preluarea a unor pasaje (piese de creaţie de tip text) dintr-o operă autentică publicată, care sunt rezumate ale unor opere anterioare operei autentice, fără precizarea întinderii şi menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
3.	Preluarea identică a unor figuri (piese de creaţie de tip grafic) dintr-o operă autentică publicată, fără menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
4.	Preluarea identică a unor tabele (piese de creaţie de tip structură de informaţie) dintr-o operă autentică publicată, fără menţionarea provenienţei şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
5.	Republicarea unei opere anterioare publicate, prin includerea unui nou autor sau de noi autori fără contribuţie explicită în lista de autori	
6.	Republicarea unei opere anterioare publicate, prin excluderea unui autor sau a unor autori din lista iniţială de autori.	
7.	Preluarea identică de pasaje (piese de creaţie) dintr-o operă autentică publicată, fără precizarea întinderii şi menţionarea provenienţei, fără nici o intervenţie personală care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	✓
8.	Preluarea identică de figuri sau reprezentări grafice (piese de creaţie de tip grafic) dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
9.	Preluarea identică de tabele (piese de creaţie de tip structură de informaţie) dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
10.	Preluarea identică a unor fragmente de demonstraţie sau de deducere a unor relaţii matematice care nu se justifică în regăsirea unei relaţii matematice finale necesare aplicării efective dintr-o operă autentică publicată, fără menţionarea provenienţei, fără nici o intervenţie care să justifice exemplificarea sau critica prin aportul creator al autorului care preia şi însuşirea acestora într-o lucrare ulterioară celei autentice.	
11.	Preluarea identică a textului (piese de creaţie de tip text) unei lucrări publicate anterior sau simultan, cu acelaşi titlu sau cu titlu similar, de un acelaşi autor / un acelaşi grup de autori în publicaţii sau edituri diferite.	
12.	Preluarea identică de pasaje (piese de creaţie de tip text) ale unui cuvânt înainte sau ale unei prefeţe care se referă la două opere, diferite, publicate în două momente diferite de timp.	

Notă:

a) Prin „provenienţă" se înţelege informaţia din care se pot identifica cel puţin numele autorului / autorilor, titlul operei, anul apariţiei.

b) Plagiatul este definit prin textul legii[1].

> *„…plagiatul – expunerea într-o operă scrisă sau o comunicare orală, inclusiv în format electronic, a unor texte, idei, demonstraţii, date, ipoteze, teorii, rezultate ori metode ştiinţifice extrase din opere scrise, inclusiv în format electronic, ale altor autori, fără a menţiona acest lucru şi fără a face trimitere la operele originale…".*

Tehnic, plagiatul are la bază conceptul de **piesă de creaţie** care[2]:

> *„…este un element de comunicare prezentat în formă scrisă, ca text, imagine sau combinat, care posedă un subiect, o organizare sau o construcţie logică şi de argumentare care presupune nişte premise, un raţionament şi o concluzie. Piesa de creaţie presupune în mod necesar o formă de exprimare specifică unei persoane. Piesa de creaţie se poate asocia cu întreaga operă autentică sau cu o parte a acesteia…"*

cu care se poate face identificarea operei plagiate sau suspicionate de plagiat[3]:

> *„…O operă de creaţie se găseşte în poziţia de operă plagiată sau operă suspicionată de plagiat în raport cu o altă operă considerată autentică dacă:*
> *i) Cele două opere tratează acelaşi subiect sau subiecte înrudite.*
> *ii) Opera autentică a fost făcută publică anterior operei suspicionate.*
> *iii) Cele două opere conţin piese de creaţie identificabile comune care posedă, fiecare în parte, un subiect şi o formă de prezentare bine definită.*
> *iv) Pentru piesele de creaţie comune, adică prezente în opera autentică şi în opera suspicionată, nu există o menţionare explicită a provenienţei. Menţionarea provenienţei se face printr-o citare care permite identificarea piesei de creaţie preluate din opera autentică.*
> *v) Simpla menţionare a titlului unei opere autentice într-un capitol de bibliografie sau similar acestuia fără delimitarea întinderii preluării nu este de natură să evite punerea în discuţie a suspiciunii de plagiat.*
> *vi) Piesele de creaţie preluate din opera autentică se utilizează la construcţii realizate prin juxtapunere fără ca acestea să fie tratate de autorul operei suspicionate prin poziţia sa explicită.*
> *vii) In opera suspicionată se identifică un fir sau mai multe fire logice de argumentare şi tratare care leagă aceleaşi premise cu aceleaşi concluzii ca în opera autentică…"*

---

[1] Legea nr. 206/2004 privind buna conduită în cercetarea ştiinţifică, dezvoltarea tehnologică şi inovare, publicată în Monitorul Oficial al României, Partea I, nr. 505 din 4 iunie 2004

[2] ISOC, D. *Ghid de acţiune împotriva plagiatului: bună-conduită, prevenire, combatere*. Cluj-Napoca: Ecou Transilvan, 2012.

[3] ISOC, D. *Prevenitor de plagiat*. Cluj-Napoca: Ecou Transilvan, 2014.

THE USING OF THE POKA-YOKE TECHNIQUES IN ORDER TO AVOID THE TRANSLATION PROGRAMS AND MENU FROM OTHER LANGUAGES

Cristian VINTILA, Carol SCHNAKOVSZKY

University of Bacau

Abstract: Poka-yoke is a quality assurance technique developed by Japanese manufacturing engineer Shigeo Shingo. The aim of poka-yoke is to eliminate defects in a product by preventing or correcting mistakes as early as possible. Poka-yoke has been used most frequently in manufacturing environments. The currently develops its Common Desktop Environment software to run in twelve locales or languages. Traditional testing of this localized software is technically difficult and time-consuming. By introducing poka-yoke (mistake-proofing) into our software process, we have been able to prevent literally hundreds of software localization defects from reaching our customers. This paper describes the poka-yoke quality approach in general, as well as our particular use of the technique in our localization efforts. Poka-yoke is providing a simple, robust and painless way for us to detect defects early in our localization efforts.

1. INTRODUCTION

Poka-yoke (pronounced "POH-kah YOH-kay") [1] was invented by Shigeo Shingo in the 1960s. The term "poka-yoke" comes from the Japanese words "poka" (inadvertent mistake) and "yoke" (prevent) [2] The essential idea of poka-yoke is to design your process so that mistakes are impossible or at least easily detected and corrected.

Shigeo Shingo was a leading proponent of statistical process control in Japanese manufacturing in the 1950s, but became frustrated with the statistical approach as he realized that it would never reduce product defects to zero.

2. CATEGORIES OF POKA-YOKE DEVICES

Poka-yoke devices fall into two major categories: prevention and detection. A prevention device engineers the process so that it is impossible to make a mistake at all. A classic example of a prevention device is the design of a 3.5 inch computer diskette. The diskette is carefully engineered to be slightly asymmetrical so that it will not fit into the disk drive in any orientation other than the correct one. Prevention devices remove the need to correct a mistake, since the user cannot make the mistake in the first place.

A detection device signals the user when a mistake has been made, so that the user can quickly correct the problem. We are surrounded every day by both detection and prevention poka-yoke devices, though we may not usually think of them as such. The microwave will not work if the door is open (a prevention device). The car beeps if I leave the key in the ignition (a detection device). At few years ago, some cars were designed not to start until the passengers had buckled their seat belts (a prevention device); but this mechanism was too intrusive and was replaced by a warning beep (a detection device).

2. CHARACTERISTICS OF GOOD POKA-YOKE DEVICES

Good poka-yoke devices, regardless of their implementation, share many common characteristics [5]:

- they are simple and cheap. If they are too complicated or expensive, their use will not be cost-effective.
- they are part of the process, implementing what Shingo calls "100%" inspection.
- they are placed close to where the mistakes occur, providing quick feedback to the workers so that the mistakes can be corrected.

3. POKA-YOKE AND SOFTWARE QUALITY

Being mainly a manufacturing technique, poka-yoke has only rarely been mentioned in connection with software development, but the philosophy behind poka-yoke has never been far from the heart of software quality. Gordon Schulmeyer [6] and James Tierney [7] refer to poka-yoke explicitly, but many software quality authors have championed detection and prevention methods in software. In 1990, Boris Beizer wrote in Software Testing Techniques [8]: "We are human and there will be bugs. To the extent that quality assurance fails at its primary purpose -- bug prevention -- it must achieve a secondary goal of bug detection". In 1993, Steve Maguire echoed a similar sentiment in Writing Solid Code [9] "All of the techniques and guidelines are the result of programmers asking themselves two questions:

- How could I have automatically detected this bug?
- How could I have prevented this bug?

3.1. Prevention devices in software

From a poka-yoke perspective, the development of computer languages could be viewed as a prevention device, since one objective of these languages is to prevent us from creating code that can be error-prone. High level languages prevent self-modifying code. Structured programming rescues us from spaghetti code. Object-oriented programming keeps us from stepping on each other's data.

3.2. Detection devices in software

Software testing is a form of detection device, but traditional system testing occurs too late in the process to allow quick, corrective feedback on mistakes. Unit testing and "smoke testing" [7] come closer to the notion of poka-yoke, in that they are located close to the source of the potential mistakes and the quick feedback they provide can keep mistakes from moving further along in the process.

The tools in software that most closely resemble poka-yoke devices are the programs such as lint, printfck, cchk, clash [10] that examine the syntax of programs and alert the programmer to a possible mistake in need of correction. Static analysis utilities are simple and cheap to run; they aim to eliminate certain classes of common mistakes; and they concentrate on the syntax of the program rather than the program's function.

Some of recent work in localizing software applications has illuminated areas that yield more readily to a poka-yoke approach than to traditional testing. The sections that follow describe our application of poka-yoke principles to solve a problem that defied a traditional software testing approach.

4. POKA-YOKE AND LOCALIZATION

To create compliant software that runs in multiple locales, developers store locale-specific strings in files called message catalogs. Rather than hard-code a text string into the application, a developer stores the text string in the message catalog and references it by its message set and message number. A message set and message number uniquely identify any message string in the catalog. Localization is the process of creating a message catalog for a particular language.

Localization is typically done after the development of the software has stabilized, and it is typically done by people external to the core development organization. These people, called localizers, receive the application's message catalog from the development organization. They then translate each message string into its equivalent expression in the target language. Localizing a software application is a difficult job. The localizers may be unfamiliar with the application. They may be located halfway around the world from the development organization. They may not even be familiar with programming. Usually, therefore, the localizer performs