

**Decizie de indexare a faptei de plagiat la poziția  
00366 / 22.03.2017  
și pentru admitere la publicare în volum tipărit**

care se bazează pe:

**A. Nota de constatare și confirmare a indiciilor de plagiat prin fișa suspiciunii inclusă în decizie.**

Fișa suspiciunii de plagiat / Sheet of plagiarism's suspicion	
Opera suspicionată (OS) Suspicious work	Opera autentică (OA) Authentic work
OS	VINTILĂ, Cristian, and SCHNAKOVSKY, Carol. The using of the POKA-YOKE techniques in order to avoid the translation programs and menu from other languages, <i>Modelling and Optimization in the Machines Building Field (MOCM)</i> . 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. <a href="http://www.faculty.washington.edu/apurva/502/Readings/Lean/pokasoft.pdf">www.faculty.washington.edu/apurva/502/Readings/Lean/pokasoft.pdf</a> .
Incidența 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
Fișa 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 <a href="http://www.plagiate.ro">www.plagiate.ro</a>	

**Notă:** Prin „p.72:00” se înțelege paragraful care se termină la finele pag.72. Notăția „p.00:00” semnifică până la ultima pagină a capitolului curent, în întregime de la punctul inițial 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. Fișa de argumentare a calificării de plagiat alăturată, fișă care la rândul său este parte a deciziei.**

Echipa Indexului Operelor Plagiate în România

## Fișa de argumentare a calificării

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

### Notă:

a) Prin „proveniență” se înțelege informația din care se pot identifica cel puțin numele autorului / autorilor, titlul operei, anul apariției.

b) Plagiatul este definit prin textul legii<sup>1</sup>.

„...plagiatul – expunerea într-o operă scrisă sau o comunicare orală, inclusiv în format electronic, a unor texte, idei, demonstrații, date, ipoteze, teorii, rezultate ori metode științifice extrase din opere scrise, inclusiv în format electronic, ale altor autori, fără a menționa acest lucru și fără a face trimitere la operele originale...”.

Tehnic, plagiatul are la bază conceptul de **piesă de creație** care<sup>2</sup>:

„...este un element de comunicare prezentat în formă scrisă, ca text, imagine sau combinat, care posedă un subiect, o organizare sau o construcție logică și de argumentare care presupune niște premise, un raționament și o concluzie. Piesa de creație presupune în mod necesar o formă de exprimare specifică unei persoane. Piesa de creație se poate asocia cu întreaga operă autentică sau cu o parte a acesteia...”

cu care se poate face identificarea operei plagiate sau suspectate de plagiat<sup>3</sup>:

„...O operă de creație se găsește în poziția de operă plagiată sau operă suspectată de plagiat în raport cu o altă operă considerată autentică dacă:

- i) Cele două opere tratează același subiect sau subiecte înrudite.
- ii) Opera autentică a fost făcută publică anterior operei suspectate.
- iii) Cele două opere conțin piese de creație identificabile comune care posedă, fiecare în parte, un subiect și o formă de prezentare bine definită.
- iv) Pentru piesele de creație comune, adică prezente în opera autentică și în opera suspectată, nu există o menționare explicită a provenienței. Menționarea provenienței se face printr-o citare care permite identificarea piesei de creație preluate din opera autentică.
- v) Simpla menționare 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 discuție a suspiciunii de plagiat.
- vi) Piesele de creație preluate din opera autentică se utilizează la construcții realizate prin juxtapunere fără ca acestea să fie tratate de autorul operei suspectate prin poziția sa explicită.
- vii) În opera suspectată se identifică un fir sau mai multe fire logice de argumentare și tratare care leagă aceleași premise cu aceleași concluzii ca în opera autentică...”

<sup>1</sup> Legea nr. 206/2004 privind buna conduită în cercetarea științifică, dezvoltarea tehnologică și inovare, publicată în Monitorul Oficial al României, Partea I, nr. 505 din 4 iunie 2004

<sup>2</sup> ISOC, D. Ghid de acțiune împotriva plagiatului: bună-conduită, prevenire, combatere. Cluj-Napoca: Ecou Transilvan, 2012.

<sup>3</sup> ISOC, D. Prevenitor de plagiat. Cluj-Napoca: Ecou Transilvan, 2014.

# Using Poka-Yoke Techniques for Early Defect Detection

Harry Robinson

---

Paper presented at the Sixth International Conference on Software Testing Analysis and Review (STAR'97)

---

## 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.

Hewlett Packard 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.

## Poka-yoke

### *History*

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. Statistical sampling implies that some products go untested, with the result that some rate of defects would always reach the customer.

While visiting the Yamada Electric plant in 1961, Shingo was told of a problem that the factory had with one of its products. Part of the product was a small switch with two push-buttons supported by two springs.

# 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 presented in this book are the result of programmers asking themselves two questions ...

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

## *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.

## *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 Hewlett Packard's 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