

# CONCEPTUL FUNCTIONALITATILOR ORTOGONALE

**Autor (i): lector asistent Andronachi Vadim, lector superior Ciorbă Dumitru**

Universitatea Tehnică a Moldovei  
Email: [andronachi.vadim@gmail.com](mailto:andronachi.vadim@gmail.com), [diciorba@yahoo.com](mailto:diciorba@yahoo.com)

**Abstract:** *Sistemul informatic este o reflectarea a unei probleme din lumea reală în termeni informatici, ceea ce înseamnă că este o conglomerare de funcționalități, care reflectă un set de necesități extrase din problema reală. În acest caz sistemul este dependent direct de metoda de organizare a funcționalităților. Una metodă de proiectare care face parte din categoria metodelor reușite este cea obiect orientată, care se aplică pentru un spectru larg de domenii, însă și ea are un set de neajunsuri care nu rezultă ca o necompletare ci ca o utilizarea incorectă a principiilor ghidate. În continuare vor fi analizate neajunsurile și prezentate soluțiile în termeni de funcționalități ortogonale.*

**Cuvinte cheie:** *sistem informatic, metoda de proiectare, funcționalități ortogonale, șabloane de proiectare.*

## 1. Fenomenul de ortogonalitate în sistemele informatice

În baza metodei de proiectare are loc organizarea entităților și a funcționalităților din sistemul informatic. Proiectarea obiect orientată organizează entitățile sistemului informatic în termeni de clasă-obiect, conform paradigmei obiect orientate. La rândul său obiectele sistemului pot fi organizate în componente sau sub-sisteme. În cazul în care între relațiile dintre clase sau componente sunt definite greșit sistemul informatic este proiectat greșit, aceasta este unica situație când sistemul informatic este proiectat greșit? Pentru a determina aceste simptome e necesar de a utiliza caracteristicile de calitate [1], în continuare vor fi prezentate unele din cele mai importante: rigiditate, super-proiectare, super-repetare.

Aceste caracteristici definesc cât de mari sunt defectele sistemului informatic. Un sistem informatic care nu implementează aceste caracteristici este un monstru. Defectele sistemului informatic pot apărea și în cazul aplicării celei mai perfecte metode de proiectare. În cele mai multe cazuri defectele sistemului informatic apar când nu sunt urmate principiile metodei (paradigmei) de proiectare. Ca exemplu unul din principiile de proiectare obiect orientată, SRP [1] ghidează încapsularea unei singure responsabilități într-o clasă (componentă), respectiv dacă clasa (componentă) dezvoltată nu urmează acest principiu, atunci posedă un nivel înalt de rigiditate.

Orișice metodă de proiectare direcționează proiectantul/programatorul spre un sistem informatic care să corespundă standardului de calitate, însă importă mult cât de explicit și concis e acest proces, aceasta fiind teza principală.

Proiectare orientată spre funcționalități ortogonale definește un set de principii care va conduce spre o construcție arhitecturală cât mai corectă și fiabilă. Baza metodologică a proiectării orientate spre funcționalități ortogonale este preluată din paradigma obiect orientată și abordarea bazată pe componente. În lucrările efectuate în [1] și [2] este observabil fenomenul de ortogonalitate, iar proiectarea orientată spre funcționalități ortogonale sistematizează acest fenomen încercând definirea construcțiilor ortogonale bazate pe principii și concepte.

## 2. Principiile funcționalităților ortogonale

Conform cercetărilor efectuate în [3] principiile sunt definite ca ghid pentru soluționarea unei probleme însă nu garantează că odată aplicate soluția obținută este fiabilă și fără defecte. Principiile sunt definite în contextul paradigmei, cu toate că pot exista și înafara paradigmei în cazul când paradigma nu e formată, sau e invizibilă. Abordarea sistemică definește noțiunea de construcție, ca fiind rezultatul aplicării principiilor și a cunoașterii asupra problemei care are a fi soluționată. Acesta este proiecția asupra principiilor prin prisma abordării sistemice.

Principiile care au a fi expuse sunt obținute prin sinteza fenomenului de ortogonalitate și aplicarea abordării sistemice:

1) Principiul profeției – la baza acestui principiu e viitorul funcționalității care are a fi implementată, ca exemplu pentru ca o clasă să implementeze careva funcționalități este necesar de a cunoaște atât contextul de utilizare cât și comportamentul așteptat și multe alte caracteristici, principiul ghidează dezvoltatorul de a privi viitorul cu câteva etape. Aplicarea corectă a acestui principiu duce la eliminarea erorilor la finele etapei, direcționează dezvoltatorul spre una din soluțiile cele mai bune. Acest principiu poate fi găsit în următoarele lucrări [1], [4] și explicit este utilizat de programarea prin testare TDD(Test Driven Development).

2) Principiul proiecției – acest principiu constă în interpretarea conceptului din diferite aspecte, diferite nivele de abstractizare, principiul ghidează alegerea corectă a responsabilității care are a fi încapsulată într-o clasă/componentă și conduce spre un nivel înalt de ortogonalitate. Acest principiu poate fi găsit în lucrările [3], [6]. De acest principiu se bucură foarte mult diagramele UML, deoarece fiecare entitate posedă comportament și atât sistemul la general cât și componentele și obiectele sistemului, respectiv putem aplica diagrama de secvențe atât pentru sisteme, componente și obiecte. Altă aplicare a acestui principiu persistă și în lucrarea [2] unde este utilizat la nivel arhitectural pentru organizarea arhitecturii în concerne.

Principiile anterioare sunt complimentarea principiilor paradigmei obiect orientate și SOLID [1]. Aplicând aceste principii în tranziția artefactelor de la etapa de analiză la etapa de programare, arhitectura sistemului informatic obține o construcție în termeni ortogonali.

### 3. Ortogonalitate și proiectare obiect orientată

Esența proiectării obiect orientate constă în dezvoltarea claselor independente și reutilizabile, ca rezultat între proiectarea orientată spre funcționalități ortogonale și proiectarea obiect orientată nu trebuie să existe ambiguități și conflicte, aceasta este obținut prin analiza principiilor și a construcțiilor obiect orientate.

O responsabilitate este ortogonală, această afirmare este găsită în [1], aplicând acest principiu asupra unei clase/componente asigurăm faptul că o clasă/componentă posedă un grad ridicat de ortogonalitate.

Dependența de abstracțiuni și de clase/componente de la nivel mai înalt garantează că clasa/componenta poate funcționa pentru mai multe implementări concrete, astfel clasa fiind ortogonală față de implementările concrete.

### 4. Ortogonalitate și șabloane de proiectare

Șabloanele de proiectare conform esenței și intenției organizează sistemul informatic în componente, fiind o complementare a proiectării obiect orientate deoarece nu posedă explicit construcții asemenea componentelor. Astfel încât șabloanele de proiectare, implementează implicit principiile de proiectare obiect orientate, ele sunt privite ca un mecanism de crearea a construcțiilor ortogonale.

### 5. Concluzie

Conform structurii evoluției științifice paradigma nu poate fi înlocuită în doi timpi cu alta, paradigma poate evolua în alta. Lucrarea dată este complementare a proiectării obiect orientate. Sistemele organizate în termeni ortogonale posedă un nivel de calitate înalt, ceea ce sa observat în multe lucrări și cercetări.

### 6. Bibliografie

1. Martin C. Robert, Martin Micah, *Agile Principles, Patterns, and Practices in C#*, Prentice Hall, July 20, 2006.
2. Nick Rozanski, Eoin Woods, *SOFTWARE SYSTEMS ARCHITECTURE, Working with Stakeholders Using Viewpoints and Perspectives*, Addison-Wesley, 2005.
3. И.В.Блауберг, *Становление и сущность системного подхода*, Наука, 1972.
4. Krzysztof Cwalina, Brad Abrams, *Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition)*, Addison-Wesley, Oct 31, 2008.
5. E.Gamma, R.Helm, R.Johnson, J.Vlissides, *Design Patterns: Elements of Reusable Object Oriented Software*, Addison-Wesley, 1995.
6. Г.Буч, Д.Рамбо, А.Джекобсон, *UML. Руководство пользователя*, ДМК, 2001.