Reglementări legale europene vs cod – cum începe construirea unor aplicații software critice în ecosistemul digital european și cum aplicăm principiile securității informației în acest proces

Octavia Lojniță

Senior Business Analyst – PMP, CSPO

Fondatoare și Vicepreședintă Women4Cyber România

1. De ce contează această temă?

• Echipa de implementare vs Echipa de business: fundamente IT, juridice și de securitate

• Probleme în proiecte critice: cod necontrolat, echipe multiple, nerespectarea reglementărilor de acces la informații clasificate/ infrastructura informatică

• Reglementările UE sunt planuri de arhitectură, nu simple documente

2. Provocări reale: Cod vs reglementare

- Lipsă de trasabilitate: Cine? Ce? Când a creat/modificat codul?
- Cod livrat neoficial de contractori
- Conflicte între echipe
- Argument: fragmente de cod fără criptare conform reglementărilor

3. Principiile fundamentale ale securității informației

Confidențialitate – informația este accesibilă doar persoanelor autorizate Ex: criptarea logurilor, acces RBAC în aplicații

Integritate – protecția împotriva modificării neautorizate a datelor
Ex: hash/checksum, versionare cod, control commit

Disponibilitate – sistemele și informațiile sunt accesibile când este nevoie
Ex: redundanță, backup-uri, SLA

Autentificare – identificarea corectă a utilizatorilor/sistemelor
Ex: MFA, autentificare bazată pe certificate

Trasabilitate – fiecare acțiune poate fi urmărită
Ex: audit trail, istoric modificări, loguri semnate

Controlul accesului – acțiuni permise doar celor autorizați
Ex: politici de acces, segregare de roluri, reguli de commit

4. Transpunerea legislației UE prin BABOK – Business Analysis Body of Knowledge

Abordarea în analiza de business:

- Punct de pornire: Reglementări UE (ex: CRA, Screening, NIS2)
- Etapa 1 – Analiză juridică: Identific articolele relevante
- Etapa 2 – Analiză de impact în sistem: Ce actor/flux este vizat
- Etapa 3 – Formularea cerințelor
  - BR – Business Requirement: ce trebuie să respecte sistemul conform legii
  - SHR – Stakeholder Requirement: cum se aplică în flux
  - FR/NFR – Functional/Non Functional Requirement: cum se traduce în cod și testare

Toate aceste elemente sunt parte din BAP (Business Analysis Package), ca livrare structurată de componente analitice(cerințe de funcționalități, scenarii de utilizare, specificații tehnice, dicționare, alte anexe)

Trasabilitate completă: articol de lege → REQ → cod → testare → livrare

5. Studiu aplicat: Transpunerea reglementărilor UE în cerințe

- **Aplicare BABOK:** reglementare juridică → BR–SHR–FR

Exemplu: Cyber Resilience Act (CRA), Art. 5 – Cerințe esențiale de securitate pentru produsele cu elemente digitale:

- BR: Aplicația trebuie să respecte cerințele CRA privind proiectarea securizată a produsului.
- SHR: Toate evenimentele legate de autentificare și acces trebuie jurnalizate în timp real.

  Jurnalul de audit trebuie stocat în format criptat, accesibil doar rolurilor autorizate.
- FR: Evenimentele logate trebuie să fie criptate AES-256 și protejate împotriva modificării.

Exemplu: Legislație europeană de reglementare sisteme informatice critice.

6. Guvernanța codului și securitatea

- Repo intern unic, commit controlat
- Automatizări pentru pachete pre-release
- Review și sincronizări regulate Dev–BA–Sec
- Scripturi pentru respingerea codului nevalidat

7. Rolul BIP: Cod extern → flux intern controlat

- BIP (Build Input Package) este bariera de control între livrarea de cod extern (contractori) și integrarea oficială în ecosistemul intern.

🔐 Ce controlează BIP înainte ca codul să ajungă în repo-ul oficial:

- Verifică dacă cerințele legale sunt respectate (ex: logare, criptare, autentificare)
- Asigură trasabilitate între cerințe și cod (ex: fiecare linie de cod legată de un REQ)
- Definește versiunea acceptată și pachetul exact care va fi integrat
- Include testare și validare în pre-producție

☑ Codul este sincronizat în repo-ul intern doar după validarea BIP.

❌ Fără BIP, codul extern rămâne în afara controlului și expune sistemul la riscuri.

Flux Git + BIP + Livrare internă

9. Activitate Contractor vs Activitate Echipă Internă

- Scenariu: același fișier modificat de 2 echipe
- Versiunea livrată conținea API key hardcoded

- Soluție: repo-uri separate, acces diferențiat, validare juridică/ echipă internă

10. Ce se întâmplă dacă NU aplicăm aceste principii?

- Cod neconform → amenzi, blocaje în implementare
- Lipsă ownership → responsabilitate neclară
- Cod livrat din repo-uri neoficiale

Câte linii de cod respectă reglementările în aplicațiile voastre?

11. Concluzii

- Reglementările UE → cerințe trasabile
- BABOK și Abordarea în analiza de business
- Business Analysis Package
- Codul = activ controlat
- Build Input Package (BIP)

- Contractorii trebuie integrați în fluxul oficial
- Dev, BA, Sec colaborează din faza de design

- Securitatea nu înseamnă doar testele de final, ci este un element de arhitectură.

VĂ MULȚUMESC!

?