

Tehnici Avansate de Programare

Fluxul de lucru cu Git și principiile SOLID

Petru Rebeja, Marius Apetrii

19 Martie 2020

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

Introdurre

Link-ul către spațiul Slack dedicat cursului va fi dezactivat azi.

Recapitulare

- Garbage collector
- IDisposable

Despre ce vom discuta azi

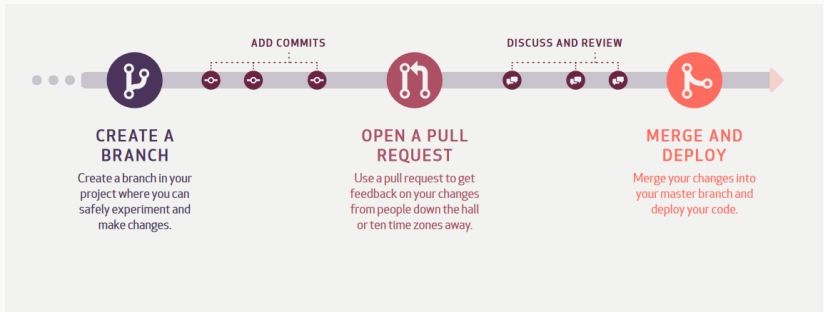
- Fluxul de lucru Git
- Ecosistemul NuGet
- Principiile SOLID

Fluxul de lucru Git



¹<https://xkcd.com/1597/>

Fluxul de lucru GitHub²



²<https://guides.github.com/introduction/flow/>

Pentru proiectele la care aveți drepturi să faceți modificări:

- Creați o ramură nouă,
- Implementați cerințele prin **modificări multiple și atomice**,
- Creați un **Pull Request** din ramura nouă,
- Revizuiți și modificați dacă este cazul,
- Îmbinați ramura nouă cu `master`,
- Livrați funcționalitatea adăugată.

Pentru proiectele la care nu aveți dreptul să faceți modificări:

- Creați un fork al proiectului,
- Adăugați modificările necesare în fork-ul propriu (folosind fluxul de lucru de mai sus),
- Creați un Pull Request din fork-ul propriu,
- Adăugați modificări suplimentare dacă este cazul
- O persoană desemnată va decide dacă modificările vor fi acceptate sau nu.

³<https://guides.github.com/activities/forking/>

Descrierea modificărilor⁴



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

⁴<https://xkcd.com/1296/>

Ghid pentru descrierea modificărilor⁵

- Lăsați o linie goală între sumar și restul descrierii.
- Sumarul trebuie să aibă maxim 50 caractere.
- Scrieți conform regulilor de gramatică și ortografie:
 - Începeți propozițiile cu majusculă,
 - Nu puneți punct după sumar.
- Descrierea trebuie să explice **ce** și **de ce** a fost modificat; nu *cum*.

⁵<https://chris.beams.io/posts/git-commit/>

Ecosistemul NuGet

Ce este NuGet?

- Aplicație utilitară pentru integrarea bibliotecilor oferite de terți în soluție/proiect.
- Facilitează instalarea pachetelor și a pachetelor de care acestea depind.
- Facilitează ștergerea și actualizarea pachetelor.

Ce este un pachet NuGet?

- O arhivă cu extensia schimbată din `.zip` în `.nupkg`.
- Conține bibliotecile necesare (fișierele `.dll`) și metadate.

- Cele publice se găsesc pe <https://www.nuget.org/>.
- Pentru pachetele private sunt necesare servere dedicate.

Principiile SOLID

- S Single Responsibility Principle
- O Open-Closed Principle
- L Liskov Substitution Principle
- I Interface Segregation Principle
- D Dependency Inversion Principle

Vrem să simulăm tranzacții financiare modelând diverse tipuri de conturi bancare (Debit, Credit etc.) precum și operațiunile care se fac asupra acestora.

Mai multe detalii în cursul II.

Single Responsibility Principle

Fiecare modul trebuie să fie responsabil pentru un singur aspect legat de funcționalitatea oferită de sistemul software. Mai mult, acel aspect trebuie să fie încapsulat în întregime de modulul responsabil⁶.

⁶https://en.wikipedia.org/wiki/Single_responsibility_principle

Open-Closed Principle

Părțile componente ale unui sistem software trebuie să fie ușor de extins dar greu de modificat⁷.

⁷https://en.wikipedia.org/wiki/Open/closed_principle

Liskov Substitution Principle

Liskov Substitution Principle

Obiectele unui sistem software trebuie să fie substituibile de către instanțe ale unor subtipuri de obiecte, fără ca substituția să afecteze corectitudinea sistemului⁸.

⁸https://en.wikipedia.org/wiki/Liskov_substitution_principle

Interface Segregation Principle

Interface Segregation Principle

Interfețele trebuie să fie mici și specifice contextului de utilizare; nu mari și generale⁹.

⁹https://en.wikipedia.org/wiki/Interface_segregation_principle

Dependency Inversion Principle

Dependency Inversion Principle

Modulele unui sistem software trebuie să depindă de reprezentări abstracte și nu de implementări concrete¹⁰.

¹⁰https://en.wikipedia.org/wiki/Dependency_inversion_principle

Înceiere

Recapitulare — fluxul de lucru Git

- O ramură nouă pentru o cerință (branch per feature)
- Modificări atomice
- Sincronizări frecvente
- Pull-Request și îmbinare în master

Este indicat să folosim NuGet pentru adăugarea de biblioteci în soluție/proiect.

- Single Responsibility Principle
- Open-Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle