Sviluppo software in gruppi di lavoro complessi[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Anno accademico 2024/25, II semestre

Lezione I: Introduzione

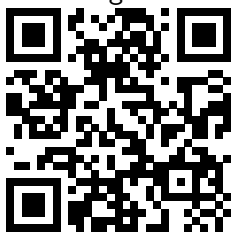
Sviluppo software in gruppi di lavoro complessi

“Svigruppo”

- II semestre, Martedì 11:30–13:30 (Aula 602), Giovedì 10:30–12:30 (Aula V3).
- proff. Monga e Bellettini

Sito: https://mameli.docenti.di.unimi.it/svigruppo
Gruppo Telegram: https://t.me/+uOF4ej4tzddkOWZk

Telegram:



- Esame: prova in laboratorio + orale

Gli obiettivi

Il corso (6CFU) presenta gli aspetti piú organizzativi dello sviluppo software, ma mira a formare figure professionali che contribuiscono allo **sviluppo**, piú che alla pura gestione dei progetti.

1. Cenni ai modelli organizzativi di sviluppo (‘cattedrale’, ‘bazaar’, ‘kibbutz’,. . . gruppi di lavoro *agili*)
2. Il supporto fornito dai tool di *configuration management* e *versioning*
3. *Continuous integration & delivery*
4. *DevOps*
5. Documentazione e specifica mirata al lavoro collaborativo (*Design By Contract* e linguaggi per la *separation of concern*)

Organizzeremo alcuni sessioni pratiche (Git, Gradle, CI, Eiffel), ma non è facile trovare le aule, forse è più facile in modalità “Bring Your Own Device”.

Sondaggio

https://u.garr.it/NLIkv

Il curriculum dello sviluppatore laureato

1. Programmazione, Algoritmi: 1 programmatore, 1 macchina (generalmente $\approx$ Von Neumann)
2. Sistemi operativi: 1 programmatore, 1 ecosistema
3. Ingegneria del software: 1 committente, 1 gruppo di lavoro, 1 sistema da realizzare e manutenere
4. Sviluppo in gruppi di lavoro complessi: *n* committenti, team eterogenei, sistemi e componenti da realizzare e manutenere.

Fare ordine nel caos!

Un caso. . .

Violet, the project manager, describes to you how her team works: "... to start with, our business representatives, together with our requirements experts, discuss with the customer what the system should do, and then write it down in the form of a requirements document. The business representatives also agree on deadlines and prices and stipulate a contract. In our case, we will be paid every time we deliver something to the customer, e.g. documents, code, or working functionality. The requirements document is then passed to our team of analysts which write the analysis document. The document, when ready, is passed to the design team, which creates the design document. I then personally distribute to the development team the various tasks, consisting of the functionalities to implement. Developers also take care of deploying the code to a test server so that it can be tested before going in production. The testers, who form a team on their own, write tests for all the functionalities that appear in the requirements document. Each test consists in a sequence of steps providing an interaction between a user and the system, and the expected result. We are planning to be done with the requirements document in two months. After that we should be done with the analysis document in one month, and it should take another month for the design document. Therefore, after four months we will start developing code. According with our estimates we should be done with the implementation in seven months. After another month devoted to testing

Risposte

https://bellettini.di.unimi.it/multiclick

1. The customer seems to be involved throughout the whole project lifecycle.
2. All the documents to be delivered act as levels of indirection between the actual requirements and the code that is supposed to implement them, increasing the likelihood of misunderstandings.
3. The development team appears to be self-organized, in the sense that developers decide wich tasks to pick for themselves.
4. Violet's testing team seems to apply test-driven development.
5. Violet's teams appear to be able to maintain a sustainable pace.
6. Violet's teams run the risk of realizing late (e.g. during the implementation) that certain assumptions made in the requirements, analysis, and/or design documents are not correct, with consequent delays in the project schedule.