

Università degli Studi di Trento

**Ingegneria del Software 2 (145924)**

**A.a. 2017 / 2018.**

**Docente: Prof. ADOLFO VILLAFIORITA**

**1. Obiettivi formativi e contenuti dell’insegnamento.**

Il corso ha l'obiettivo di trasferire agli studenti alcune pratiche correnti di gestione e di sviluppo software, consentendo agli studenti di sperimentarle sul campo attraverso la realizzazione di un piccolo progetto di sviluppo.

Il corso è organizzato in tre parti:

* La prima parte del corso riprende i concetti di base di gestione progetti e di sviluppo dei progetti software, presenta l’archittetura delle moderne applicazioni web e introduce gli strumenti utilizzati per lo sviluppo del progetto.
* Nella parte centrale del corso gli studenti, organizzati in gruppi di 2-3 persone, metteranno in pratica le tecniche e i concetti presentati nella prima parte del corso. Verranno inoltre identificate alcune milestone, in corrispondenza delle quali i gruppi di lavoro dovranno consegnare specifici prodotti di progetto. I prodotti consegnati saranno oggetto di valutazione e concorreranno alla determinazione del voto finale.
* Nella parte finale del corso si terranno alcune “retrospettive” di progetto, in modo da evidenziare le “lesson learned” e, se necessario, riprendere e consolidare alcuni dei concetti sviluppati in maniera meno completa o sistematica.

**2. Risultati attesi.**

Al termine del percorso, lo studente sarà in grado di:

* maneggiare tecniche e strumenti utilizzati per organizzare e gestire lo sviluppo di sistemi software, con particolare riferimento allo sviluppo di applicazioni web
* Essere in grado di impostare un progetto di sviluppo software, dal concept al rilascio, con tecniche tradizionali e agili
* Conoscere bene le seguenti attività di ingegneria del software: project costing e pricing; specifica sistemi con user stories e con specifiche eseguibili (e.g., BDD, cucumber); cenni a test-driven development; configuration management e sistemi di versionamento distribuiti; code inspection, code review e metriche di qualità; continuous integration and deployment

**3. Metodi didattici e attività di apprendimento.**

L’insegnamento adotta una pluralità di metodi didattici:

* **Lezioni frontali** sono utilizzate per presentare le tecniche e le metodologie utilizzate per lo sviluppo di sistemi software e che verranno utilizzate dai gruppi di lavoro durante lo sviluppo del progetto.
* La parte centrale del corso è costituita da un **progetto di sviluppo software**, in cui gli studenti, **organizzati in piccoli gruppi,** sperimentano le tecniche e gli strumenti presentati durante il corso. Per il progetto, il docente organizzerà gruppi di lavoro di due/tre persone, tenendo conto delle proposte degli studenti. Ciascun gruppo si impegna a svolgere le consegne che verranno assegnate, nella forma di milestone di progetto. Le consegne saranno oggetto di valutazione e concorreranno a determinare il punteggio di uscita del corso.
* Nella parte finale del corso o in corrispondenza di consegne significative verranno proposte delle “retrospettive di progetto”, in cui **verrà chiesto agli studenti di esporre e discutere** le principali difficoltà emerse durante lo svolgimento delle consegne e come queste siano state affrontate; spazio verrà inoltre dedicato ad una discussione sui vantaggi e sulle difficoltà incontrate nell’applicazione delle tecniche illustrate a lezione.

**4. Conoscenze di base.**

L’efficace apprendimento e il pieno conseguimento dei risultati attesi sono facilitati dalla padronanza dei metodi e delle tecniche di sviluppo e ingegnerizzazione del software presentati nel corso Ingegneria del Software I. La padronanza della lingua inglese semplifica la comprensione dei materiali e dei lucidi (tutti in inglese).

**5. Lingua.**

L’insegnamento è impartito in lingua italiana. Tutto (o quasi tutto) il materiale didattico è disponibile esclusivamente in lingua inglese.

**6. Frequenza.**

La frequenza assidua alle lezioni frontali e la partecipazione attiva durante il corso costituiscono elementi centrali per l’apprendimento.

La partecipazione al progetto di sviluppo software è un elemento integrante per l’apprendimento delle tecniche presentate al corso e consente ai frequentanti di accumulare punti validi per la valutazione finale.

Gli studenti non frequentanti integreranno le conoscenze con letture aggiuntive (vedi materiali e programma) e con una prova scritta, sostitutiva del progetto.

**7. Materiali didattici.**

***Studenti frequentanti.***

* Materiale presentato a lezione (ad es., lucidi) che verrà reso disponibile sul sito web del corso.
* Sam Ruby and Dave Thomas. 2016. *Agile Web Development with Rails 5*. Pragmatic Bookshelf.
* Adolfo Villafiorita. 2014. *Introduction to Software Project Management*, CRC Press

***Studenti non frequentanti.***

* Materiale presentato a lezione (ad es., lucidi) che verrà reso disponibile sul sito web del corso.
* Sam Ruby and Dave Thomas. 2016. *Agile Web Development with Rails 5*. Pragmatic Bookshelf.
* Adolfo Villafiorita. 2014. *Introduction to Software Project Management*, CRC Press

## Matt Wynne e Aslak Hellesoy, 2017, [*The Cucumber Book: Behaviour-Driven Development for Testers and Developers*](https://www.amazon.it/Cucumber-Book-Behaviour-Driven-Development-Developers-ebook/dp/B06XR5S98S/ref=sr_1_5?ie=UTF8&qid=1501062284&sr=8-5&keywords=cucumber+ruby)*,* Pragmatic Bookshelf

**8. Verifica dell’apprendimento.**

L’apprendimento e il raggiungimento dei risultati attesi sono verificati con le seguenti modalità.

**Studenti frequentanti.**Il voto finale per gli studenti frequentanti è espresso in 30esimi, con voto massimo pari a 32/30esimi. Punteggi superiori a 30 possono determinare la decisione del docente di assegnare la lode. La lode viene assegnata discrezionalmente, come elemento distintivo di eccellenza del lavoro svolto.

**La verifica per i frequentanti è determinata dallo svolgimento del progetto.** Più specificamente: nella fase di impostazione del progetto verranno identificate dal professore delle milestone, in corrispondenza delle quali verrà chiesto agli studenti di produrre determinati prodotti (ad es., documenti, mockup, codice), ad ognuno dei quali è assegnato un punteggio in 30esimi (ad es., 5 punti su 30). Puntualità e qualità dei prodotti consegnati determineranno il punteggio assegnato ad ogni consegna. I voti delle consegne verranno poi sommati fino a determinare un voto massimo di 30/30esimi. I voti conseguiti dal gruppo di lavoro valgono per tutti i membri del team.

Per i **non frequentanti è prevista invece una prova scritta,** con domande aperte e chiuse. La prova, nella modalità closed-book, consente di acquisire 32 punti. Nello scritto è specificato, per ogni domanda, il punteggio massimo assegnabile. Lo scritto è composto da un numero variabile di domande, tipicamente compreso tra 5 e 10.

**La prova scritta può essere svolta anche dagli studenti frequentanti che non fossero soddisfatti del voto conseguito con il progetto.** La partecipazione alla prova scritta annulla il voto conseguito nel progetto.

**9. Orari d’insegnamento e aula.**

Come indicato nel programma analitico.

**10. Orari di ricevimento.**

Il ricevimento studenti per l’insegnamento è previsto **in un giorno della settimana che verrà concordato con gli studenti a inizio corso.** I ricevimenti si tengono presso l’ufficio del docente, previo appuntamento, ottenibili inviando una mail di richiesta almeno il giorno prima.

**11. Contatti**

**Docente: Adolfo Villafiorita**

e-mail: [adolfo.villafiorita@fbk.eu](mailto:adolfo.villafiorita@fbk.eu)

Ufficio: Fondazione Bruno Kessler  
 via Sommarive 18

<http://www.ict4g.org/home/reach_us.html>