

La metodologia aziendale Agile applicata al Coding

Domenico Consoli

Istituto di Istruzione Superiore “Manfredi Tanari”
Viale Felsina 40, 40139 Bologna
domenico.consoli@manfreditanariedu.it

Abstract. In questo articolo si vuole applicare la metodologia aziendale Agile alla realizzazione di un’applicazione (videogame) condivisa in una classe seconda di un Istituto tecnico Commerciale che sarà sviluppata con il linguaggio Scratch. La metodologia Agile si applica già nello sviluppo del software soprattutto quando i requisiti richiesti dai clienti sono oggetto di continui cambiamenti e il team di sviluppo, periodicamente, rilascia diverse release. Un videogioco è un ambiente complesso e per la sua progettazione e realizzazione sono richieste diverse competenze a partire dal disegno dello sfondo (stage) fino alla scelta degli attori o avatar principali e alle interazioni tra di loro e/o con l’ambiente circostante. Questa metodologia si basa su una didattica laboratoriale che mette al centro il processo di apprendimento degli studenti che lavorano a piccoli gruppi e portano avanti dei compiti specifici. La metodologia Agile stimola di più lo studente a sviluppare, con maggiore coinvolgimento, entusiasmo e responsabilità, l’applicazione. Durante l’intero processo di sviluppo gli studenti rivestiranno i diversi ruoli aziendali tra cui quello di clienti e si riuniranno spesso per apportare, in maniera incrementale, dei miglioramenti al prodotto/applicazione.

Keywords: Agile, Scrum, Coding.

1 Introduzione

Il progetto si sta sviluppando in una classe seconda dell’ Istituto di Istruzione Superiore “Manfredi Tanari” ad indirizzo Amministrazione Finanza e Marketing (AFM). In questa classe, per la prima volta, si è introdotta la logica del pensiero computazionale e del coding.

Oramai siamo entrati, a pieno titolo, nell’era della società digitale, delle Smart cities, dei Big Data, del Cloud Computing e dell’ IoT (Internet of Things). I dispositivi diventano sempre più intelligenti e possono comunicare e scambiare dati tra di loro. Le applicazioni digitali “invadono” sempre di più tutti gli spazi: casa, scuola, palestre, uffici e fabbrica. Non basta più che i cittadini e in particolare gli studenti sappiano utilizzare questi strumenti; occorre che in tutti gli ordinamenti scolastici si studino e sperimentino i primi approcci con la teoria della programmazione e gli algoritmi. I giovani nativi digitali, più che all’utilizzo dei dispositivi elettronici, devono concentrarsi sullo sviluppo del software, il cosiddetto coding per apportare, in

futuro, il loro contributo nella costruzione di una società digitale sempre più attenta al benessere della comunità [1].

Per realizzare questo obiettivo si è pensato di assegnare alla classe il compito di realizzare un videogame.

La realizzazione di un videogame corrisponde ad un'esperienza progettuale che integra diverse competenze, sia tecniche (sviluppo del coding) che trasversali (artistiche, musicali, narrative, scenografiche,...). In un videogioco si possono suscitare negli studenti delle emozioni: paure o soddisfazioni. Gli studenti imparano a vincere la paura mettendo in campo delle azioni per combattere i nemici in contesti ostili e nello stesso tempo possono provare delle soddisfazioni quando vincono e acquisiscono dei punti.

Oggi, all'interno delle aziende, nello sviluppo di determinati progetti/prodotti e soprattutto nello sviluppo del software si utilizza una metodologia Agile e un framework più rigido ad essa associata Scrum. Oramai da diversi anni si è sperimentato e verificato che questa metodologia aumenta la produttività aziendale, la qualità del prodotto/servizio e riduce i tempi di consegna al cliente il quale viene maggiormente coinvolto nel processo di sviluppo.

L'articolo è così strutturato: nel prossimo paragrafo si descrive, con maggiori dettagli, il framework Agile/Scrum e dopo si passa alla discussione dell'esperienza didattica di realizzazione del videogame applicando i vari passaggi della metodologia aziendale. Alla fine si trae qualche conclusione.

2 La metodologia Agile e il framework Scrum

Nell'ambito del management aziendale il metodo Agile [2] si contrappone al tradizionale approccio a cascata (Waterfall) che prevede la suddivisione del ciclo di vita di un progetto/prodotto in schemi rigidi e sequenziali. Il metodo Agile è più flessibile e dinamico e privilegia gli individui e le interazioni piuttosto che i processi e gli strumenti, la collaborazione col cliente più che la negoziazione dei contratti ed è predisposto al cambiamento piuttosto che seguire un piano ben predefinito.

L'idea del Metodo Agile non si basa sull'approccio classico e lineare di progettazione, ma sulla possibilità di realizzare un progetto/prodotto per fasi/riunioni, chiamate "sprint". Ad ogni sprint corrisponde una nuova funzionalità e viene verificata la soddisfazione del cliente, al quale viene mostrato il lavoro svolto fino a quel momento. Questo sistema iterativo (ed interattivo) consente di apportare facilmente delle modifiche al prodotto, di abbattere i costi di produzione e, soprattutto, di evitare sforzi inutili ed un eventuale fallimento del progetto.

Una tecnica che implementa la metodologia Agile è Scrum [3] che opera in contesti turbolenti e variabili.

Scrum è un framework che consente alle persone di risolvere problemi complessi di tipo adattivo e al tempo stesso rilascia prodotti di altissimo valore in maniera efficace e creativa. Questo framework mette in discussione il project management classico che si basa sulla scomposizione delle attività e dei processi in maniera deterministica. Con Scrum si è più tempestivi nel soddisfare le richieste dei clienti e nell'offerta di un prodotto personalizzato. Esso si basa sui concetti del kaizen e della lean thinking:

breve pianificazione, piccoli investimenti, intensa collaborazione, un procedimento per piccoli passi, tutte azioni che consentono di correggere gli errori strada facendo e con una comunicazione più efficace tra tutti gli stakeholder.

Durante le fasi di progetto e degli sprint c'è un continuo confronto tra commitment (cliente) e il team di development per approvare step by step l'evoluzione delle attività. Pianificando in modo incrementale, si prevede una documentazione del progetto/prodotto più leggera.

Come il pensiero computazionale anche Scrum si basa su tre concetti: creatività, collaborazione e approccio iterativo e incrementale. Entrambi riconoscono il ruolo della creatività come veicolo di espressione e di propensione al miglioramento continuo.

Entrambi esaltano le persone, la comunicazione tra di loro e il lavoro di gruppo come strumenti per raggiungere obiettivi più grandi di quelli che il singolo potrebbe raggiungere da solo. Con un approccio iterativo e incrementale si può suddividere un problema complesso in altri più semplici. La soluzione dei problemi più semplici per tentativi e errori (approccio empirico) permette di arrivare alla soluzione del problema più complesso originario.

La filosofia di Scrum si basa su persone che svolgono certi ruoli aziendali e che si incontrano periodicamente anche con il commitment per apportare delle migliorie incrementali al prodotto/servizio.

Scrum riconosce 3 ruoli aziendali fondamentali (Fig. 1):

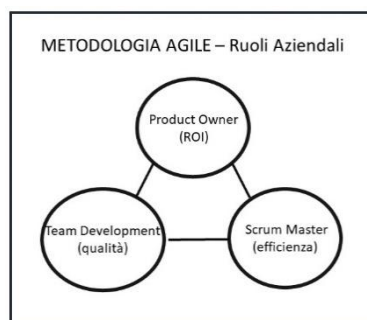


Fig.1 I ruoli aziendali nella metodologia Agile

Product Owner. E' la persona che conosce tutti i requisiti del prodotto e che si interfaccia con il team e con i clienti e in generale con tutti gli stakeholder. E' colui che massimizza il ROI (Return of Investment) (*redditività*) e il valore del prodotto e del lavoro svolto.

Scrum Master. E' il responsabile del processo che deve accertarsi che il team lavori in maniera coerente con lo sviluppo del progetto eliminando tutti i vari ostacoli che possano interporsi (*efficienza*).

Team development. Si occupa dello sviluppo del prodotto e del testing delle funzionalità (*qualità*), e ha la responsabilità di organizzare le priorità delle attività per portare a termine un determinato sprint.

In Scrum si possono organizzare 4 eventi/cerimonie/riunioni (Fig. 2). Ogni riunione viene identificata con il termine *sprint* e si pianifica per ogni fase di realizzazione del prodotto:

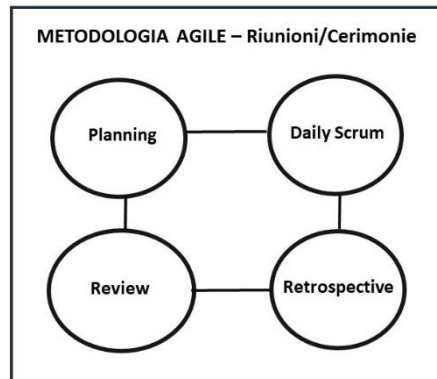


Fig.2. Le cerimonie nella metodologia Agile

Planning. Il Product Owner che ha stilato il Product Backlog, in presenza del Team di Sviluppo e dello Scrum Master, descrive gli item più importanti e l'obiettivo da raggiungere negli sprint successivi.

Daily Scrum. Un confronto giornaliero (della durata di 10-15 min) fra Team di Sviluppo e Scrum Master, il quale annota il lavoro svolto il giorno precedente e crea un piano per il prossimo sprint.

Review. Una revisione alla fine di ogni sprint per valutare se l'obiettivo prefissato è stato raggiunto e con quali risultati. Partecipa a questo sprint anche il cliente (committente) del prodotto, al quale verrà mostrato il lavoro svolto fino a quella release.

Retrospective. A questa riunione partecipa tutto il team development per valutare il processo e vedere cosa ha funzionato, cosa no e cosa bisogna cambiare.

Scrum prende in considerazione anche degli *artefatti* (Fig. 3) che servono a visualizzare i progressi e cioè tutti gli incrementi del prodotto/servizio.

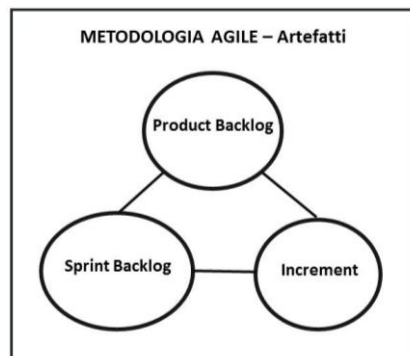


Fig.3. Gli artefatti nella metodologia Agile

Product backlog. Documento che contiene la lista di tutti i requisiti, per ordini di priorità, necessari per la realizzazione del prodotto e viene gestito dal Product Owner.

Sprint backlog. Tutti i task da completare nel singolo sprint e vengono prelevati dal Backlog.

Increment. Tutti gli elementi del backlog che sono stati completati durante uno sprint e quelli precedenti.

3 Metodologia didattica nella costruzione del videogame con Scratch

L'idea è stata quella di portare avanti un'esperienza di apprendimento e una didattica laboratoriale in modo da coinvolgere tutti gli studenti di una classe, suddivisi per gruppi, nella realizzazione di un'applicazione [4]. Una didattica che mira allo sviluppo delle competenze di un gruppo-studenti non può prescindere da un approccio di tipo laboratoriale, una metodologia che riconosce e valorizza il ruolo attivo degli alunni nei processi di problem solving e di attivazione di un pensiero critico e riflessivo.

In una didattica laboratoriale si mettono in atto diverse metodologie: cooperative learning [5], lavoro di gruppo, discussione e aiuto reciproco. In questo modo si stimolano negli studenti le abilità collaborative, la motivazione, il confronto, l'aiuto reciproco e il rispetto dell'altro.

Nel laboratorio gli studenti riflettono e condividono informazioni/idee per la soluzione di un problema reale e la realizzazione di un progetto che viene affrontato da diversi punti di vista (varie discipline). La competenza da acquisire diventa il risultato di una elaborazione pratica, di una riflessione e di una interiorizzazione del processo di apprendimento che si concretizza in un prodotto fisico e/o concettuale. Il laboratorio trasforma le conoscenze e abilità apprese dall'insegnamento dei docenti e li rende utilizzabili in contesti diversi.

Il framework Scrum, descritto nel paragrafo precedente, si focalizza sulla teoria del processo di apprendimento che si basa su 3 fasi: segui la regola/rompi la regola/ricostruisci la regola.

Nel "*seguire la regola*" lo studente segue scrupolosamente gli insegnamenti del maestro/docente. Anche se esistono diversi approcci si concentra solo sul modo che ha appreso dal docente. In "*rompi la regola*" lo studente comincia ad ampliare i suoi orizzonti, ad apprendere da altri docenti, a sperimentare altre pratiche e a seguire in autonomia altri percorsi. Nel "*ricostruire la regola*" lo studente incomincia a imparare dalla sua pratica. Crea il suo personale modo di affrontare le situazioni e adatta ciò che ha appreso alle particolari circostanze.

Il progetto che devono portare avanti gli studenti in laboratorio è la realizzazione di un videogame e come termine di consegna della release finale del videogioco è stato dato loro il 30 maggio. Quindi il lavoro che devono sviluppare è ancora in progress.

La realizzazione di un videogame si presta bene ad una suddivisione dei compiti a gruppi di studenti appartenenti alla stessa classe. Un gruppo si occuperà della sceno-

grafia (stage o palcoscenico), un altro gruppo della creazione dei personaggi che si prenderanno in considerazione, un altro dei testi, delle musiche e così via. Quindi ad ogni gruppo sono demandate certe competenze.

Il compito assegnato al gruppo studente va definito, studiato e contestualizzato in maniera critica e costruttiva (problem posing) [6] tale da trovare delle soluzioni adeguate (problem solving) [7].

Per la composizione dei gruppi è stato chiesto loro, inizialmente, di farlo in autonomia e le poche persone che sono rimaste fuori sono state assegnate dal docente (Product Owner) per cercare di uniformare la composizione dei gruppi in base alle loro abilità e predisposizioni individuali.

Nella ricerca della soluzione, all'interno di un gruppo, si stabilisce un dialogo alla pari, peer-to-peer e quindi si possono sperimentare delle forme di apprendimento dialogico [8] dove è importante la forza delle idee e delle proposte dei singoli che lavorano allo stesso livello e senza alcuna prevalenza gerarchica.

Nel gruppo l'apprendimento cooperativo [9] valorizza il learning by doing e il learning by experience. Le persone che acquisiscono certe competenze nella risoluzione di un problema saranno in grado di risolverne altri. Gli studenti, in questo modo, diventano più consapevoli, più coinvolti e più responsabili e propositivi. L'eterogeneità del gruppo come i momenti di riflessione individuali e collettivi [10] sono dei fattori importanti che combattono i comportamenti spontanei a volte dispersivi.

Nella creazione di un videogioco servono delle competenze informatiche che prendono in considerazione tutte le strutture fondamentali di un linguaggio di programmazione: start/end, relazioni sequenziali di tipo temporali (prima-dopo) e di causa-effetto, test decisionali (se.... allora.... altrimenti), cicli iterativi (ripeti finchè...) e così via.

Queste strutture di tipo logico-matematiche emergono quando si stabiliscono le regole e la meccanica del gioco: struttura *if...then...else* (se non prendi l'arco o la spada non puoi abbattere il nemico), struttura *iterativa* (ripeti i lanci della freccia in continuazione) e così via.

Le abilità che gli studenti devono possedere e migliorare sono diverse e vanno dalla logica, dal pensiero critico, dalla creatività alle espressioni letterarie, artistiche e musicali. Inoltre nella co-progettazione ludica subentreranno anche delle emozioni e dei sentimenti [11].

Il compito assegnato agli studenti favorisce la socialità e l'espressione personale in un ambiente stimolante e divertente nel quale l'errore non viene visto come un fallimento ma come uno spazio di prova dove condividere le proprie idee e emozioni.

Gli alunni in questo modo acquisiranno delle abilità quali l'autoconsapevolezza, la creatività ed espressione artistica, la narratività e la capacità di mettersi in gioco con gli altri.

In questa esperienza gli studenti sono liberi di ideare e progettare le singole scene, gli attori, le musiche e questo dà spazio alla loro immaginazione e creatività.

Questo modo di lavorare può essere visto anche come una metodologia di studio e apprendimento della disciplina Informatica che utilizza la realizzazione di un videogioco come strumento didattico ed educativo. In questo modo gli studenti diventano soggetti attivi che possono esprimere la loro creatività e progettualità in un contesto

di scuola partecipativa dove Scratch rappresenta lo strumento per creare i vari soggetti/oggetti.

Il videogame prima di diventare “opera” e prodotto finale è una pratica, un'esperienza, una concatenazione di atti concreti fisici e affettivi. A lavorare a stretto contatto con i compagni è entusiasmante e si possono trasmettere vari sentimenti. E' un momento in cui si pone l'attenzione sulla propria affettività, creatività e sul lavoro di squadra. Lo studente può esprimere la sua personalità in evoluzione, mettendosi in relazione con gli altri e dando vita ad azioni in cui ciascuno trova un proprio spazio. Nella comunicazione e nel confronto con gli altri si può sviluppare la propria auto-consapevolezza e le fondamenta per un agire responsabile, creativo, cooperativo e non violento.

La costruzione di un videogioco li spinge anche ad andare oltre al compito principale che hanno ricevuto: immaginare e progettare nuovi livelli di gioco, pensare e realizzare il packaging della scatola, disegnare il logo del contenitore e così via.

4 Applicazione della metodologia Agile/Scrum al lavoro di gruppo

Le regole base della metodologia Agile mirano a stabilire una più netta interazione tra le risorse umane coinvolte nel progetto, indipendentemente dai processi da seguire e dagli strumenti utilizzati. Infatti le persone si riuniscono frequentemente in sprint per discutere, vedere lo stato di avanzamento dei lavori, interpretare gli altri requisiti richiesti dal commitment, apportare gli aggiornamenti e rilasciare nuove release del prodotto/servizio. Uno dei principali obiettivi è quello di preferire la consegna (rapida) di un prodotto/servizio quale può essere un'applicazione, un software effettivamente funzionante ad una documentazione spesso sproporzionata per dimensioni e contenuti.

Un'altra priorità della metodologia Agile è quella di soddisfare il cliente e in particolare di rilasciare un software di qualità e funzionante che può essere aggiornato velocemente al cambiare dei requisiti richiesti. E' previsto un ruolo centrale per il Committente, il quale è chiamato a collaborare con il team di programmatori in diverse fasi o sprint.

Il team di lavoro riflette, nei vari sprint, su come diventare sempre più efficiente ed efficace e quindi regola e adatta di conseguenza il suo comportamento.

Nella metodologia Agile si seguono diversi obiettivi: reazione efficace (rapida e adattiva) ai cambiamenti, comunicazione efficace tra tutti gli stakeholder, coinvolgimento del cliente nel team di sviluppo, stretta collaborazione tra stakeholder di business e team di sviluppo e frequente rilascio di nuovo software funzionante.

Il metodo Agile, pianificando in modo incrementale, prevede una documentazione più leggera e non la stesura di una grande mole di dati.

Nello sviluppo del videogame gli studenti vengono suddivisi in gruppi di 4 persone e in certi sprint si aggiunge una quinta persona che rappresenta il cliente o committente. Questo committente partecipa a tutti gli sprint review e a turno segue gli sprint dei vari gruppi. Anche questo ruolo svolto da uno studente è importante perché in questo modo l'alunno apprende come interfacciarsi con il development team, come formulare i requisiti e come misurare il soddisfacimento di questi requisiti tramite le release

fornite.

Inoltre a turno uno dei quattro studenti di ogni gruppo svolge il ruolo di Scrum Master e quindi coordina il lavoro del development team.

Mediamente gli studenti fanno uno sprint di 15 minuti dopo una lezione di due ore. Il primo sprint Planning è il più duraturo di tutti (5 ore) è quello in cui dopo che lo studente-committente ha descritto i requisiti del videogioco richiesto, il docente (Product Owner) stila la lista delle attività da svolgere (Product Backlog). Lo Scrum Master e tutti gli studenti del development team procedono alla scelta del tipo di videogioco da realizzare, tutte le tematiche da portare avanti, i punteggi (incrementi/decrementi) e così via.

Scelto il tipo di videogioco, si passa alla scelta degli attori/avatar e alle azioni da eseguire. Quindi si danno delle indicazioni sul palcoscenico (stage) e sulle interazioni tra i vari personaggi.

In questo sprint si analizzano le performance e si procede a illustrazioni cartacee che sintetizzano il videogame.

Successivamente allo sprint Planning si fanno degli sprint regolari giornalieri di 15 minuti.

Nei vari sprint ogni studente si pone delle domande: cosa dovevo fare?, cosa ho fatto? e cosa farò per il prossimo sprint?.

Per ogni sprint il development team e lo Scrum Master elaborano il piano di test da mettere in campo in presenza del commitment per validare o meno la release che è stata prodotta.

Durante lo sprint si prendono in considerazione una parte di item (attività da svolgere) dall'elenco completo del Product Backlog la cui stesura è stata curata dal Product Owner (Docente).

Dopo il secondo sprint in presenza dello studente-commitment si fa un'analisi delle attività eseguite e si mostra al cliente il lavoro svolto fino a quel momento. Nello sprint review è importante la presenza del cliente per mostrare e fare una demo del prodotto e accettare e rifiutare quella particolare release o sottoporla a specifiche modifiche.

Dopo 5 sprint e cioè 5 lezioni di due ore si fa una retrospettiva. Nella retrospettiva dei singoli gruppi, dove saranno presenti il development team e il PO, si analizzano i processi per vedere le cose da cambiare totalmente, cosa modificare e cosa invece lasciare.

All'ultima retrospettiva partecipano tutti i gruppi per una discussione generale sulla release del videogioco finale.

In questo sprint di retrospettiva si consegna la release del videogioco finale.

Se si prende in considerazione la preparazione dei vari stage o palcoscenici il gruppo che si deve occupare di questo e quindi il development team in uno sprint visiona tutta la galleria degli stage che si trovano in scratch. Si selezionano quelli che possono essere presi in considerazione per il videogioco che si deve realizzare. Una volta che si è selezionato lo stage si possono apportare tutte le modifiche che interessano con la barra degli strumenti dell'editor grafico che è incorporato in Scratch. Lo stage può essere importato da file esterni di tipo *.jpg, *.gif, *.png,... oppure addirittura scattando una immagine con la webcam. Quindi durante gli sprint e anche le review che il development team fa insieme al commitment si può benissimo prendere un book esterno di immagini e selezionare quelle che si desiderano o che si avvicina-

no ai requisiti richiesti dal commitment.

Dopo dal linguaggio scratch si può passare da uno sfondo all' altro utilizzando l'istruzione/mattoncino "passa allo sfondo seguente".

In fase di retrospettiva il team e lo Scrum Master può riflettere sui processi di creazione dei vari stage e cambiare anche modalità. Se le attività eseguite non sono soddisfacenti si può rimediare creando prima degli schizzi su carta e dopo elaborare lo sfondo con un programma grafico professionale tipo Photoshop o Adobe Illustrator e dopo salvarlo come file grafico e importarlo dentro Scratch oppure in maniera meno professionale si può creare lo stage dall' inizio utilizzando la barra dell' editor grafico incorporato in Scratch.

Altre riflessioni possono essere portate avanti durante i vari sprint, review o retrospettive con gli altri gruppi che devono curare i testi, le musiche o devono disegnare degli attori protagonisti del videogioco.

5 Conclusioni

Negli ultimi anni con la globalizzazione e l'innovazione digitale spinta, i modelli di gestione aziendale si sono dovuti adattare alle nuove condizioni "turbolente" di produzione, caratterizzate da altissima competizione e richieste di maggiore produttività, velocità e qualità.

La metodologia AGILE e il framework SCRUM supportano l'organizzazione aziendale nella gestione di "progetti complessi" grazie all'utilizzo di un sistema iterativo (ed interattivo) che consente di apportare velocemente modifiche al progetto/prodotto e il rilascio di frequenti release del prodotto o software con aggiornamenti incrementali.

La metodologia AGILE valorizza le persone, le interazioni, la collaborazione e il cambiamento. Il confronto continuo tra il development team e il committente-cliente favorisce lo sviluppo di un prodotto/progetto più affidabile e di maggiore qualità. Con questa metodologia possono trarre dei benefici sia il team di sviluppo che il management e l' azienda.

Per il team ci sarà una maggiore responsabilizzazione, un maggiore coinvolgimento e una maggiore efficienza. Per il management un allineamento più diretto con il business, una panoramica, step by step, dello stato del progetto/prodotto, un monitoraggio più efficiente e una più semplice e immediata risoluzione di problemi. Per l'azienda c'è il vantaggio di migliorare la relazione e la soddisfazione dei clienti, una maggiore velocità di esecuzione e produttività.

Dal punto di vista didattico è importante utilizzare questa metodologia per cambiare la modalità di insegnamento dell' Informatica.

A scuola generalmente l' informatica si spiega sempre allo stesso modo (metodologia Waterfall o a cascata: analisi, progettazione, codifica e test. Anche il docente nel fare la sua lezione utilizza un metodo a cascata: lezione frontale/esercizio/compiti e verifica. A scuola lo sviluppo di un programma è sempre visto come un lavoro individuale e non come un lavoro di gruppo. Quindi gli studenti sono sempre abituati a sviluppare un programma da soli e quando si trovano a lavorare in azienda dove lo sviluppo di un' applicazione complessa richiede l'apporto di più professionisti non si trovano a

loro agio.

La metodologia AGILE può invece stimolare una nuova didattica che si basa sul lavoro di gruppo e sulla gestione delle risorse e delle responsabilità condivise.

Il framework Scrum utilizza diverse strategie volte a minimizzare i passaggi burocratici tra gli sviluppatori e favorisce l' interazione interna (team) ed esterna (clienti) alternando frequentemente le fasi di progettazione e codifica. In questo modo si è più efficiente e proattivi.

Lo sviluppo di un' applicazione, in base alla metodologia Agile, offre molti vantaggi in termini di produttività e di qualità del prodotto finale.

Il videogioco che gli studenti devono sviluppare può essere visto come uno strumento espressivo ed educativo e come un approccio didattico, una metodologia in grado di raggiungere, in maniera efficace, risultati di apprendimento in tutte le discipline scolastiche per le molteplici forme espressive che coinvolge (scenografie, attori, testi, musiche, ...). Nella realizzazione del videogioco si innescano forme di apprendimento cooperativo e collaborativo. Il videogioco aiuta a sviluppare le capacità individuali e di gruppo degli studenti e li stimola ad una crescita sociale e relazionale perché favorisce il lavoro di gruppo e dà spazio a diverse forme di immaginazione, creatività e anche a mettere in campo delle specifiche abilità nella ricerca degli algoritmi per il problem solving e il decision making.

References

1. Consoli, D: I nativi digitali e il coding. In Proc. DIDAMATICA 2015 - Dalla Società della Conoscenza alla Società delle Competenze, pp. 1-8, Genova (2015)
2. Lisca, F.: Il quinto paradigma. Come trasformare la propria azienda in un'organizzazione agile, Franco Angeli, 2017
3. Schwaber, K, Beedle, M., Agile software development with Scrum, Prentice Hall, 2002
4. Agile School Manifesto, <https://www.infoq.com/articles/agile-schools-education>, last accessed 2/04/2019
5. Doderò, G., Gennari, R., Melonio, A., Torello, S.: Gamified Co-design with Cooperative Learning, In Proc. CHI 2014 Extended Abstract on Human Factors in Computing Systems, pp. 707-718, New York (2014)
6. English L. D., Fox J. L. and Watters J. J., Problem posing and solving with mathematical modeling. *Teaching Children Mathematics*, 12, 3, 2005, 156-163.
7. D'Amore, B.: *Problemi, Pedagogia e psicologia della matematica nell'attività di problem solving*, ed. Franco Angeli, Milano, 1996
8. Flecha, R.: *Sharing words: theory and practice of dialogic learning*, Rowman & Littlefield Publishers, US, 2000
9. Kaye, A.: *Apprendimento collaborativo basato sul computer*, *Tecnologie Didattiche* n. 4, 1994
10. Levy P. *Collective Intelligence: Mankind's Emerging World in Cyberspace*, Perseus, Cambridge, 1997
11. Brondino, M., Doderò, G., Gennari, R., Pasini, M., Raccanello, D., Torello, S.: Emotions and Inclusion in Co-Design at School: Let's Measure Them!, In Proc. *Methods and Intelligent System for Technology Enhanced Learning*, pp. 1-8, Springer, Cam (2015).