

Didattica per progetti a confronto: meglio Scrum o Waterfall?

Marcello Missiroli, Daniel Russo, and Paolo Ciancarini

DISI, Università di Bologna
Mura Anteo Zamboni 7, 40127. Bologna
{marcello.missiroli,daniel.russo,paolo.ciancarini}@unibo.it

Abstract. Un esperimento su base nazionale mette a confronto due metodologie di sviluppo di programmi radicalmente opposte. La scelta influenza significativamente il prodotto degli studenti, favorendo aspetti funzionali nel caso di Scrum e quelli non funzionali nel caso Waterfall. Non traspare un decisivo vantaggio didattico a favore dell'uno o l'altro metodo, evidenziando come la scelta dipenda da fattori quali la struttura della classe e la metodologia didattica del docente. Si nota il vantaggio della metodologia Project Based Learning in generale e su ipotizza una “terza via” come possibile sintesi.

Keywords: Project Based Learning, Agile, Waterfall

1 Introduzione

All'inizio del secolo la metodologia Agile [2] fa il suo ingresso sulla scena dello sviluppo del software, proponendo un approccio radicalmente diverso rispetto al passato. Si tratta di un compromesso tra regole rigide gradite alle aziende e “nessuna regola” come gradito agli sviluppatori. L'approccio risulta più adattativo che predittivo, più orientato al dialogo che al seguire ciecamente un piano. Dopo quasi vent'anni il metodo Agile non è più una curiosità per pochi entusiasti ma una pratica affermata che porta notevoli vantaggi non solo nelle PMI ma anche in grandi imprese [10]. Diversi studi mostrano che la metodologia Agile è generalmente vantaggiosa, se eseguita correttamente [1].

Tuttavia, queste “nuove” metodologie sono alquanto ignorate nell'insegnamento dell'Informatica, tanto all'università quanto nelle scuole — in particolare in Europa e in Italia. Il che è tanto più strano se notiamo come diversi valori Agili risultino strettamente imparentati con i concetti tipici dei moderni approcci didattici basati sul costruzionismo e sul costruttivismo. Perfino la ricerca dedicata all'argomento è limitata, con qualche eccezione (come [3], [8]) che tuttavia non offre analisi rigorose dei risultati, nonostante della ricerca specifica è stata svolta in materia [6], [5], [4].

Abbiamo realizzato un esperimento in cui lo stesso progetto software è realizzato da due diversi team all'interno della stessa classe, ciascuno con una metodologia differente. L'esperimento è stato svolto in sette classi in varie scuole

in varie città, del Nord, Centro e Sud Italia. Questo confronto ci ha permesso di osservare il potenziale didattico dei due metodi, limitando l'effetto di fattori di disturbo quali il livello della classe, metodi di insegnamento, parametri di valutazione.

I risultati mostrano un lieve vantaggio per Agile, ma non sufficiente a raccomandarne un'adozione generalizzata. Visto l'effetto positivo sugli studenti, si consiglia comunque di insegnare almeno un metodo di sviluppo, idealmente proponendo aspetti di entrambe le strade, purché integrati in un progetto didattico coerente.

Il presente articolo di sviluppa nel seguente modo. Nella sezione 2 illustriamo la metodologia di ricerca utilizzata nella nostra esperienza. La struttura dell'esperimento è invece spiegata nella sezione 3. L'analisi dei dati raccolti verrà illustrata nella sezione 4. Successivamente, nella sezione 5 sono riportate le opinioni di studenti e docenti circa l'esperienza fatta. Infine, nella sezione 6, tireremo le nostre conclusioni ed interpretazioni, con raccomandazioni e direzioni future.

2 Metodologia

Abbiamo voluto “simulare” un ambiente lavorativo. Il nostro esperimento prevedeva, dopo una breve lezione introduttiva, una sessione unica di lavoro di 6 ore, con una breve pausa, durante la quale i ragazzi avrebbero sviluppato un progetto web di una certa complessità. Più specificamente, uno stesso progetto sarebbe stato sviluppato in modo indipendente da due gruppi all'interno della stessa classe, in modo da minimizzare gli effetti ambientali (livello della classe, programma svolto, metodologia di insegnamento) e mettere così in evidenza pregi e difetti del metodo. L'esperimento è stato eseguito in più classi in diverse parti d'Italia.

2.1 Difficoltà riscontrate

Imbastire un esperimento scientifico in una scuola pubblica non è un'impresa facile. Nel caso specifico, i punti di maggiore criticità si sono rivelati i seguenti:

- Ignoranza da parte dei docenti di Informatica, che hanno dimostrato di conoscere poco le metodologie Agili.
- Bilanciamento tra sezione informativa ed operativa in due sessioni di 2 e 6 ore complessive.
- Il carico burocratico per ottenere l'autorizzazione dell'amministrazione e riorganizzare l'orario scolastico nelle mattine interessate.
- Nell'anno considerato (2016), Informatica non era tra le materie dell'Esame di Stato (per gli ITIS), per cui la giustificazione di tale attività risultava debole.

Per i motivi indicati la ricerca di partner è risultata molto difficile. Abbiamo contattato colleghi in circa 200 istituti tramite email o telefonate dirette, poiché non esiste un elenco nazionale dei docenti di Informatica. Di questi, 20 docenti si sono dichiarati interessati ma, in definitiva, solo 7 istituti hanno davvero partecipato.

2.2 Metriche

È molto difficile stabilire il successo di un progetto, poiché il significato di “successo” dipende in grande misura del contesto. Nel caso in esame si voleva stabilire il successo di una simulazione rispetto a *tre* distinti obiettivi: costruire in un software che soddisfi le richieste, implementare realisticamente un metodo di sviluppo software, e la validità dell’esperienza come strumento didattico. Serviva quindi identificare tre diverse metriche per misurare il raggiungimento di tali obiettivi. Abbiamo scelto i seguenti:

- Progresso (**PROG**): una percentuale che indica la quantità di requisiti soddisfatti (Waterfall) o di user story realizzate (Scrum).
- Voto (**GRA**): utilizzato come una misura del successo dell’esperienza da parte del docente.
- Punteggio (**SCORE**): valuta il processo applicato al prodotto, rispondendo implicitamente alla domanda “Abbiamo seguito le regole del metodo”? La differenza con PROG risiede nel fatto che — ad esempio — una funzione realizzata durante la fase di testing incrementa il punteggio di progresso generale, ma non quella del punteggio, perché realizzata violando i dettami del metodo. Per una migliore granularità di analisi abbiamo tenuto separati i punteggi relativi a richieste funzionali e non funzionali, nonché i punteggi delle due fasi dello sviluppo.

A tutto questo abbiamo aggiunto un questionario introspettivo per i ragazzi e una intervista ai docenti al termine dell’anno scolastico.

3 Struttura dell’esperimento

Gli esperimenti sono stati eseguiti da gennaio a maggio 2016; la dimensione dei team variava da 4 a 6 (la media era comunque 5), per un totale di circa 160 studenti partecipanti. L’esperimento includeva due parti, una “informativa” e una “operativa”, seguita da una serie di operazioni conclusive. Per limitare i problemi di interpretazione è stato preparato un documento di presentazione inviato a tutti i docenti partecipanti.

3.1 Sezione informativa

Consiste di una esperienza di 2 ore, il cui scopo è quello di fornire agli studenti una comune base di riferimento delle conoscenze teoriche e pratiche delle due metodologie di sviluppo. Questo passo è stato ritenuto necessario perché i metodi Agili non sono parte del curriculum standard; inoltre il metodo Waterfall è insegnato solitamente solo in teoria (e rapidamente dimenticato).

L’esperienza aveva la seguente struttura:

- *Lezione Waterfall*. Una presentazione di mezz’ora sullo sviluppo Waterfall. Mostra le varie fasi, ruoli, gli artefatti principali, vantaggi e svantaggi.

- *Laboratorio Waterfall*. La classe lavora in piccolo gruppi, ognuno dei quali riceve il testo di un Esame di Stato di qualche anno prima. Hanno quindi 15 minuti per delineare la struttura generale di un documento dei requisiti (in gergo, SRS) e dettagliare almeno un elemento (di solito, il diagramma E-R).
- *Lezione Scrum*. Una presentazione tradizionale di mezz’ora sullo sviluppo Agile in generale e Scrum in particolare. Anche in questo caso si definiscono i termini, i ruoli, vantaggi e svantaggi.
- *Laboratorio Scrum*. La classe lavora in piccolo gruppi, ognuno dei quali riceve il testo di un Esame di Stato diverso dal precedente. Hanno quindi 20 minuti per scrivere tre User Story, di cui almeno una con Condizioni di Accettazione, nonché selezionare le proprie “Definition of Done” da una lista di 6 oggetti.
- *Conclusioni*. Tutto il materiale prodotto viene raccolto e inviato al docente per commenti (senza voto). L’insegnante usa gli ultimi 5 minuti a disposizione per delineare come sarà organizzato l’esperimento.

Le lezioni sono state effettuate in parte da uno degli autori, in parte dai docenti della classe e in un caso tramite un video — soluzione adottata per limitare i costi di trasferta.

3.2 Sezione operativa

La parte più importante dell’esperimento richiedeva una esperienza laboratoriale di 6 ore continuate. La sequenza delle operazioni è stata la seguente:

- Si informa gli studenti della formazione dei gruppi; ogni gruppo è accoppiato (a caso) ad un altro.
- Ad ogni coppia è assegnato un progetto, e per ogni coppia si estrae chi lo svolgerà con Waterfall e chi con Scrum. Ora i gruppi sono pronti per partire, ed questo punto il cammino dei due gruppi diverge:
 - Agli sviluppatori Waterfall si fornisce una notevole quantità di materiale: descrizione generale del progetto, una lista di richieste in formato MoSCoW, un SRS completo di circa 30 pagine con casi d’uso, diagrammi ER, interfacce utente, specifiche funzionali e non funzionali. È loro concesso un primo periodo di 5 ore per la codifica del progetto, al termine del quale ci si attende un codebase “feature-freeze”. Dopo di che, il team ha un’ulteriore ora per il testing, senza sviluppare nuove funzioni. Durante l’esperimento, l’insegnante assume il ruolo di un manager non tecnico. Pertanto, dovrà astenersi dal dare consigli tecnici, e offrire invece consigli di tipo manageriale (problemi di tempo, comportamento, ...)
 - Agli sviluppatori Scrum è dato molto meno materiale: una descrizione generale del progetto; *definition of done*, user story ripartite in tre categorie di priorità. Il tempo a disposizione in due mini-sprint di 2 ore e mezza. Ognuna consiste della Sprint preparation (15’), esecuzione (110’), consegna e retrospect (5’). Una pausa obbligatoria è inserita tra i due sprint. In questo caso, l’insegnante assume il ruolo di Product Owner del team e come tale potrà fornire indicazioni sul prodotto finale, ma non gli

sarà premesso dare ordini o criticare il lavoro altrui. Dovranno inoltre stabilire se la storia è stata davvero “done, done, done”.

Al termine dell’attività tutto il materiale realizzato è stato raccolto e inviato a noi: codice, dump dei database, password, e altro.

3.3 Operazioni conclusive

Nei giorni immediatamente successivi all’esperienza, ai docenti è stato chiesto di compilare un modulo di valutazione. Ciò ha permesso di stabilire quali *feature* siano state effettivamente realizzate, il voto ricevuto dal gruppo ed una valutazione soggettiva della performance complessiva del gruppo. Parimenti, un sondaggio elettronico è stato inviato a tutti gli studenti, chiedendo la loro opinione sull’attività. Infine, verso la fine dell’anno, tutti gli insegnanti sono stati contattati di persona o telefonicamente per effettuare una intervista di riflessione sull’esperienza.

4 Analisi dei dati

4.1 Confronto diretto

Per prima cosa abbiamo confrontato i punti dei team che, all’interno di una stessa classe, lavoravano allo stesso progetto con una metodologia diversa. In questo modo limitiamo l’impatto di disturbi dovuti all’ambiente, metodi di insegnamento, preparazione della classe e simili. Il risultato è mostrato nella Tabella 1. Segnaliamo che per un disguido abbiamo solo 13 confronti diretti anziché 16; inoltre, l’usabilità è stata misurata da noi tramite un test preorganizzato.

Table 1. Confronto diretto

	Vantaggio <i>Scrum</i>		Pareggio	Vantaggio <i>Waterfall</i>	
	10+%	5%-10%	Delta <5%	5%-10%	10+%
SCORE-F	5	3	4		1
SCORE-XF	1		2		10
SCORE	3	2		2	1
PROG	2	5	2	1	3

È facile notare che i team Scrum si sono concentrati sulle richieste funzionali, mentre i team Waterfall su quelli di altro tipo. Complessivamente, si nota un lieve vantaggio per Scrum nel punteggio (SCORE), che diventa più marcato nel caso di PROG (che non tiene conto delle penalità dovute al processo). I test di usabilità sono risultati negativi in entrambi i casi. I progetti Waterfall offrivano una interfaccia piacevole e talvolta innovativa, ma le funzionalità davvero operative erano davvero minime (1 o 2 al massimo). Al contrario, i progetti Scrum offrivano un maggior numero di funzionalità spesso bloccate da una interfaccia utente grezza e che richiedeva elevate conoscenze informatiche per funzionare.

4.2 Analisi statistica

Abbiamo quindi analizzato l'insieme dei risultati per verificare se si rilevava una differenza significativa dovuta alla metodologia. Per cui, abbiamo raggruppati gli esperimenti per metrica (SCORE-F, SCORE-XF, SCORE, PROG) per ogni metodologia. Abbiamo voluto vedere se c'era una differente significatività statistiche fra le due metodologie tramite un *t Test* sulle medie (presupposte uguali).

I calcoli effettuati ci dimostrano che la nostra ipotesi è confermata solo nel caso della performance extra-funzionale, che presenta una differenza significativa, poiché la *t Stat* è in valore assoluto maggiore del coefficiente *t Critico* a due code. Da ciò concludiamo che nello specifico caso dei requisiti extra-funzionali, Waterfall è più performante di Scrum. In tutti gli altri casi, da un punto di vista puramente statistico, non è possibile affermare che un metodo risulti migliore dell'altro. Per un maggior livello di dettaglio, si invita a fare riferimento ad un nostro precedente lavoro [7].

5 Opinioni di studenti e docenti

5.1 Impressioni degli studenti

Le opinioni dei ragazzi sull'esperienza effettuata sono state raccolte tramite un questionario online. In particolare, si voleva stabilire se l'attività era risultata piacevole e/o utile. La maggior parte delle domande prevedevano una risposte con una scala Likert a 7 livelli, mentre due domande avevano una risposta aperta. Abbiamo deciso di ritenere rilevanti i risultati per i quali il risultato si discostava dal valore medio di 4 per un $\Delta \geq 0,5$.

I risultati (Tabella 2) mostrano come, in generale, l'attività sia stata ricevuta positivamente dagli studenti, che hanno lavorato bene in gruppo. Anche se non molto pronunciata, chi ha usato Scrum è stato più contento della metodologia, e non l'avrebbe cambiata volentieri. Le domande aperte mostrano che gli aspetti più graditi sono risultati la possibilità di lavorare in gruppo seguiti dall'opportunità di lavorare in gruppo, confermando i risultati delle sezioni a risposte chiuse. Tra le cose non piaciute, i più si sono lamentati dello scarso tempo per completare il progetto e le diatribe interne al gruppo.

5.2 Aspettative dei docenti

Riteniamo che il voto ricevuto dagli studenti sia un fattore decisivo, poiché quello che al momento, e più di ogni altro, misura il successo di una attività formativa. Giudicando i risultati, non si nota una grande differenza a seconda delle metodologie di sviluppo usate. Notiamo però un miglioramento nel voto medio; nel caso di Waterfall è 7,21 con un incremento sensibile sul voto medio base di riferimento (6,5). Per Scrum l'incremento è meno pronunciato ma comunque pari a circa mezzo punto. Tale risultato è visibile in tutti i team, indipendentemente dal tipo di scuola, classe, tipo di progetto (tranne in 9% dei casi).

Table 2. Sondaggio studenti

Domande	<i>WF</i>	<i>SCRUM</i>	<i>AVERAGE</i>
Ti è piaciuta l'attività?	4.92	5.29	5.11
Come ha lavorato il tuo gruppo?	5.14	5.14	5.14
Hai imparato qualcosa?	3.89	4.59	4.24
Pensi che questa metodologia sia utile nel contesto lavorativo?	4.20	4.76	4.48
Pensi che questa metodologia sarà utile per la tua carriera?	4.50	4.70	4.60
Tra le due modalità viste, quale pensi sia la più efficiente*?	4.05	4.76	4.40
Ti senti più preparato per Scrum o Waterfall*?	3.56	4.36	3.96
Avresti preferito usare l'altra metodologia?	3.56	3.09	3.33

Table 3. Aspettative docenti

	Waterfall					Scrum				
	P1	P2	P3	P4	AVG	P1	P2	P3	P4	AVG
Voto	6.83	6.58	8.33	8.50	7.21	8.08	6.19	6.80	6.50	7.07
Aspettativa	3.17	3.17	3.67	3.00	3.24	3.33	3.50	3.80	3.50	3.53
Distinta Aspettative	2 Peggio, 9 Uguale, 6 Meglio					1 Peggio, 9 Uguale, 8 Meglio, 1 Molto meglio				

Tutte questo risulta ancora più interessante se lo confrontiamo con le aspettative degli insegnanti (Tabella 3). A ciascuno di essi è stato chiesto se i risultati erano conformi alla loro aspettative, indicandoli con valore da 1 (Molto peggio) a 5 (Molto meglio); ne risulta che i risultati superano le attese in tutti i progetti, leggermente meglio nel caso di Scrum. Nel Waterfall, si hanno risultati migliori in 6 casi su 13, mentre in Scrum in ben 9 risultati su 13, di cui uno particolarmente brillante.

Naturalmente, l'interpretazione di questo dato è dubbia: o il metodo Scrum era più adatto alle caratteristiche del team, oppure gli insegnanti non sono stati in grado di stimare la performance nel caso di Scrum, che conoscevano meno. Ma il punto più importante è che tutti sono stati positivamente sorpresi dai risultati del metodo di simulazione di progetto, sia esso Scrum o Waterfall.

5.3 Opinione dei docenti

L'ultima parte della nostra ricerca consiste in una intervista strutturata ai docenti coinvolti nell'esperimento. Lo scopo era quello di stabilire se l'esperienza aveva avuto successo, e se i docenti percepivano Scrum come un'alternativa realistica per l'insegnamento dell'informatica; proprio per questo motivo le interviste sono differite di 2-4 mesi rispetto all'effettuazione dell'esperienza. Le interviste sono state effettuate di persona o per telefono quindi trascritte e analizzate.

Presentiamo qui una sintesi delle risposte, seguendo il metodo generale proposto in [9].

D1: “Come definiresti le tue conoscenze delle metodologie Agili? Come le hai acquisite?” Prima di questa attività, il 55% degli insegnanti aveva solo un’idea vaga di cosa fossero tali metodologie, e per alcuni era un argomento totalmente ignoto. Dopo l’attività, tutti hanno dichiarato di avere alcune conoscenze di base — alcuni hanno proseguito ad auto documentarsi. Alcuni hanno dichiarato un livello ‘sufficiente’ di conoscenza, acquisito in modo personale o all’università.

D2: “Le metodologie Agili come Scrum possono diventare uno strumento standard per l’insegnamento?” Il 55% degli insegnanti pensa di sì, notando che gli studenti hanno gradito Scrum un po’ di più. Altri sono più cauti, ritenendo che ogni metodo abbia i suoi pregi. Realisticamente, un docente ha affermato che in fondo si tratta di una simulazione: “Direi che non seguiamo davvero un metodo Waterfall, nel senso che gli studenti raramente seguono i passi in modo preciso poiché noi docenti, interagiamo e lavoriamo con loro. Ma non seguiamo neppure un metodo Agile, perché gli studenti non hanno una coscienza precisa di quello che stiamo facendo. È probabilmente un ibrido”.

D3: “Pensi che questo esperimento sia stato utile nell’insegnare agli studenti le basi di entrambe le metodologie?”. La stragrande maggioranza (89%) pensa di sì, seppur con qualche riserva relativa al Waterfall. Un docente segnala che “erano probabilmente più concentrati sul fare che a riflettere sul processo complessivo”.

D4: “L’esperimento era ben progettato? I tempi erano giusti?”. Il 78% si è lamentato che sarebbe occorso più tempo, sia per la preparazione sia per l’esecuzione; tuttavia, metà di loro riconosce che l’esperienza necessitava di compromessi. Nessuno ha menzionato o notato il fatto che la gestione del fattore tempo facesse parte dell’esperimento.

D5: “L’esperimento ha avuto effetti sulla classe nel lungo periodo? Soprattutto in termini di voti, motivazione, autostima...”. Due docenti hanno notato un miglioramento di entusiasmo e motivazione; altri due un atteggiamento più positivo verso l’argomento e il lavoro in gruppo. La maggioranza non ha rilevato grandi differenze.

D6: “Durante l’esecuzione, si sono attenuti al ruolo assegnato o scelto?” Non si nota una tendenza generale, anche se i gruppi Waterfall tendono a seguire più i ruoli rispetto ai gruppi Agili. Tuttavia, sembra essere un fattore che dipende dal singolo gruppo, come rileva un collega: “Qualche gruppo ha scelto il leader sbagliato, e da lì tutto è andato a rotoli”.

D7: “Qualche fatto di rilievo che val la pena riportare?”. Il 55% degli insegnanti ha notato una partecipazione positiva e intensa, anche tra studenti solitamente passivi. Due commenti sono particolarmente interessanti: “Inizialmente la classe non era tanto interessata all’esperienza, ma man mano che lavoravano ne erano sempre più coinvolti.”; “Un team non ha iniziato il progetto nel modo corretto, concentrandosi sull’interfaccia utente. Pur consigliati, hanno ostinatamente perseguito la loro strada, con risultati pessimi. Spero che questo abbia loro insegnato qualcosa”.

D8: “Commenti?” Tutti i docenti hanno trovato l’attività interessante, e circa la metà di loro pensa di riproporla l’anno venturo, magari aumentando il tempo dedicato a pianificazione e design. “Dovremmo dedicare più tempo all’osservazione degli studenti durante il lavoro, per esempio assegnando un osservatore dedicato alle dinamiche interne. Più tempo significherebbe una migliore osservazione dell’impatto della metodologia sui gruppi. “.

6 Interpretazione e conclusioni

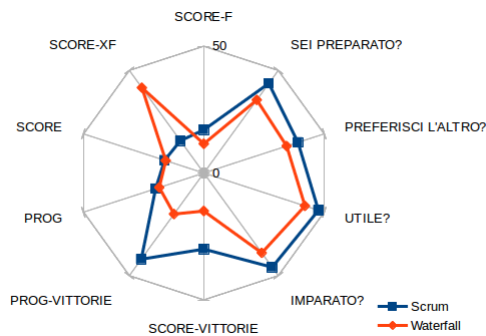


Fig. 1. Confronto dei principali risultati

I dati raccolti non si prestano ad una interpretazione univoca. I dati puramente numerici, sintetizzati nello schema Radar (Fig. 1), mostrano un deciso vantaggio di Waterfall nel campo dei requisiti non funzionali e di processo, mentre in tutti gli altri aspetti Scrum ha un vantaggio, anche se non troppo marcato.

La situazione non cambia se teniamo conto delle opinioni di studenti e docenti: un lieve vantaggio per la nuova metodologia, ma non sufficiente a consigliare un cambio così radicale di metodologia. Inoltre, Waterfall è ancora la metodologia più diffusa nel campo lavorativo; non è quindi possibile eliminarla dal curriculum — senza contare le limitate conoscenze del corpo docente sulle metodologie Agili.

Un punto comune a tutte le esperienze è l’interesse per la metodologia ispirata al *Project-Based Learning* (PBL), che ha condotto a un miglioramento generale dell’ambiente di apprendimento e spesso dei voti. Ci sentiamo quindi di suggerire un incremento dell’uso di tali metodologie didattiche, lasciando al docente la scelta del metodo esatto da seguire — alcuni docenti magari preferiranno il controllo fornito dal un progetto Waterfall, altri potranno scegliere una metodologia Agile.

Esiste però una terza possibilità — l’introduzione di una “terza via”: **Waterscrum**. Il termine definisce in modo dispregiativo il primo passo verso una completa transizione della produzione verso Scrum (o altro metodo Agile); è di una pratica diffusa, seppur non proprio ideale nel caso dell’industria. Tuttavia, nel ramo educativo, Waterscrum potrebbe essere il punto d’incontro, in grado di

gestire sia le necessità di pianificazione e progettazione per programmatori ancora inesperti sia le opportunità di creatività e libertà offerte dalle metodologie Agili. Permetterebbe ai docenti di espandere le proprie conoscenze gradualmente, senza sprecare risorse in corsi, pur consentendo un certo grado di personalizzazione dovuto alle necessità della classe e del docente. Tutto questo, ovviamente, richiede ulteriori studi e approfondimenti.

Per riassumere, possiamo dire che questo esperimento dimostra un deciso effetto positivo della metodologia PBL sull'ambiente di apprendimento, ma non può dimostrare che un cambio radicale di metodologia di insegnamento verso le metodologie Agili abbia di per sé effetti positivi - pur evidenziando diverse caratteristiche potenzialmente vantaggiose. Tuttavia, vari elementi ci fanno pensare che alcuni elementi della metodologia Agile possano essere introdotti fin dalle scuole superiori, perché la loro diffusione nel mondo del lavoro sembra essere in continuo aumento. La scelta di come e quanto "Agilizzare" il proprio corso dipenderà dalla esperienza e sensibilità del singolo docente.

Suggeriamo degli indirizzi di ricerca in ambito didattico per implementare Waterscrum, all'interno dell'alveo del PBL. Inoltre, questi nuovi curricula dovranno essere validati in classi sperimentali, per valutarne l'efficacia.

Tutti i materiali sono disponibili online all'indirizzo https://bitbucket.org/marcello_missiroli/svw-data.

References

1. Chow, T., Cao, D.B.: A survey study of critical success factors in agile software projects. *Journal of Systems and Software* 81(6), 961–971 (2008)
2. Highsmith, J., Fowler, M.: The agile manifesto. *Software Development Magazine* 9(8), 29–30 (2001)
3. Meerbaum-Salant, et al.: An agile constructionist mentoring methodology for software projects in the high school. *ACM Transactions on Computing Education* (2010)
4. Missiroli, M., Russo, D., Ciancarini, P.: Learning agile software development in high school: an investigation. In: *Proceedings of the 38th International Conference on Software Engineering Companion*. pp. 293–302. ACM (2016)
5. Missiroli, M., Russo, D., Ciancarini, P.: Una didattica agile per la programmazione. *Mondo Digitale* 15(6) (2016)
6. Missiroli, M., Russo, D., Ciancarini, P.: Agile for millennials: a comparative study. In: *International Workshop on Software Engineering Curricula for Millennials (SECM)*. IEEE (2017)
7. Missiroli, M., Russo, D., Ciancarini, P.: Teaching test-first programming: assessment and solutions. In: *Computer Software and Applications Conference (COMPSAC)*, 2017. IEEE (2017)
8. Romeike, R., Göttel, T.: Agile projects in high school computing education: emphasizing a learners' perspective. In: *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*. pp. 48–57. ACM (2012)
9. Seaman, C.B.: Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering* 25(4), 557–572 (1999)
10. The Standish Group: Chaos report 2013 (2013), <https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf>