



Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci



Approccio Agile Environmental Intelligence for Agriculture

Riferimento	
Versione	1.0
Data	27/12/2022
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Carmine Laudato, Pierluigi Lambiase
Approvato da	



Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F. Ferrucci

Revision History

Data	Versione	Descrizione	Autori
27/12/2022	1.0	Stesura Documento	Carmine Laudato Pierluigi Lambiase



Sommario

Revision History	2
1. Introduzione	4
2. Modello Agile	4
2.1. CVS Ibrido	6
2.2. La Servant Leadership	6
2.3. Scrum	6
2.4. Kanban	7
2.5. Extreme Programming	7
2.6. Mix di approcci	8
3. Motivazione scelte	8



1. Introduzione

In questo documento verranno descritti le motivazioni sulle scelte del modello di sviluppo del prodotto EnIA.

2. Modello Agile

La "metodologia agile" è un approccio allo sviluppo del software basato sulla distribuzione continua di software efficienti creati in modo rapido e iterativo.

Benché l'espressione faccia riferimento specifico a un determinato sviluppo del software, questa metodologia non prevede regole da rispettare tassativamente nell'ambito dello sviluppo del software. Si tratta piuttosto di un tipo di approccio alla collaborazione e ai flussi di lavoro fondato su una serie di valori in grado di guidare il nostro modo di procedere.

I team di sviluppo software devono tenere conto di:

- **Individui e interazioni** rispetto a processi e strumenti
- **Un software efficiente** rispetto a una documentazione esaustiva
- **La collaborazione con il cliente** rispetto alla negoziazione dei contratti
- **La preparazione ad affrontare il cambiamento** rispetto all'esecuzione di un piano

Di seguito verranno elencati i dodici principi del "Manifesto Agile":

1. Soddisfare il cliente attraverso una consegna anticipata seguita da consegne cicliche.

È importante consegnare rapidamente il prodotto finale per soddisfare le esigenze dei clienti, che sono abituati ad avere aggiornamenti regolari e non vogliono attendere a lungo per vedere i cambiamenti richiesti. Ciò è particolarmente importante nello sviluppo del software, poiché viene utilizzato per una vasta gamma di scopi.

2. Accogliere favorevolmente le mutevoli esigenze, anche in fase avanzata di sviluppo

il cambiamento è costante e le metodologie del passato per affrontare le esigenze mutevoli stanno diventando obsolete. I principi sono necessari per supportare i mercati in evoluzione, le esigenze dei clienti, la concorrenza spietata e la capacità di cambiare direzione quando necessario.

3. Fornire software funzionante con frequenza ciclica

Il principio di sviluppo agile è diventato essenziale a causa delle elevate esigenze di documentazione per la pianificazione nello sviluppo del software. Adottando questo principio, si può ridurre il tempo di pianificazione e dedicare più energie ai progetti, il che porterà il team a lavorare in modo più agile.

4. I quadri business e gli sviluppatori lavorano insieme ogni giorno

La metodologia Agile coinvolge partner, clienti e ingegneri che lavorano insieme per raggiungere gli obiettivi del progetto, riducendo i rischi legati allo sviluppo e concentrandosi sullo scambio e la



Laurea Magistrale in Informatica-Università di Salerno Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

collaborazione. Più intensamente la gestione è coinvolta nel processo, più facile sarà per loro identificare le sfide affrontate dagli ingegneri e le implicazioni delle modifiche apportate durante lo sviluppo.

5. Organizza i progetti con persone motivate

Un team agile è in grado di creare software di qualità se è composto da persone esperte e affidabili, che richiedono una quantità specifica di fiducia. Se non si può fidarsi degli ingegneri, è meglio non coinvolgerli nel progetto.

6. Trasmettere informazioni parlandosi faccia a faccia

Parlarsi faccia a faccia è uno dei migliori metodi di scambio di informazioni e offre una comprensione chiara di come l'altra persona percepisce un compito o una funzione specifici. Tuttavia, il principio non richiede che tutti i colleghi partecipino sempre alla comunicazione faccia a faccia, ma piuttosto che i partecipanti siano motivati a impostare modalità di scambio diretto e trasparenti.

7. Il software funzionante è la misura principale del progresso

Concentrarsi sul seguire un piano prestabilito può distrarre dall'aggiornamento della documentazione e distogliere l'attenzione dall'obiettivo del progetto. È importante invece fare del software il punto centrale del progetto e non perdere mai di vista l'obiettivo finale.

8. Mantieni un ritmo costante a tempo indefinito

I team di progetto possono diventare stanchi quando lavorano su un progetto per un lungo periodo di tempo, un problema comune nello sviluppo del software. La metodologia Agile propone di organizzare il lavoro in cicli brevi per evitare la stanchezza e mantenere la motivazione delle persone. I responsabili di progetto senior o gli Scrum Masters dovrebbero impostare la giusta velocità dei cicli sin dall'inizio per consentire agli ingegneri di essere più produttivi.

9. Prestare persistente attenzione all'eccellenza tecnica

Un'attenzione costante all'eccellenza tecnica e un solido piano di sviluppo migliorano l'agilità dei team di sviluppo software. Ciò consente loro di creare software funzionante e prodotti di alta qualità, riducendo la probabilità di bug e malfunzionamenti nelle modifiche al codice.

10. La semplicità, vale a dire l'arte di massimizzare la quantità di lavoro non svolto, è essenziale.

I team Agile devono percepire che c'è sempre la possibilità di migliorare e che l'innovazione offre metodi più flessibili e semplici per fare le cose. La ricerca sulla gestione dei progetti aiuta i manager e le aziende a lavorare costantemente sulla loro metodologia per massimizzare i risultati e ridurre gli sforzi. Ciò richiede uno sforzo continuo.

11. I team si auto organizzano

Questo principio promuove l'emergenza di team di talento e ispirati che hanno un insieme di competenze dinamiche, si assumono le responsabilità, discutono in modo coerente con gli altri colleghi e forniscono idee per fornire prodotti di qualità. A intervalli regolari, i team effettuano operazioni di retrospective su come migliorare l'efficienza e si auto-organizzano di conseguenza.

12. I team si analizzano retrospettivamente e ottimizzano i loro comportamenti



Laurea Magistrale in Informatica-Università di Salerno Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

la riflessione sui progressi e sugli insuccessi e la capacità di modificare rapidamente i metodi per affrontare il fallimento sono importanti per il successo della metodologia Agile. Ciò richiede comunicazione, feedback, comprensione delle strategie Agile e un ambiente che favorisca sia lo sviluppo che l'apprendimento dagli insuccessi. Anche se nessun team funziona sempre con successo, un team responsabile, istruito e capace può lavorare su sé stesso adottando le misure giuste per migliorare ulteriormente i propri progressi.

2.1. CVS Ibrido

Il ciclo di vita seguito, durante la fase di design, è il modello waterfall con retroazione. Per la fase di implementazione verrà utilizzato l'approccio Agile "Scrum" con qualche elemento di Kanban ed Extreme Programming (XP).

Utilizzando gli approcci Agile, è possibile ottenere i deliverables quanto prima e questo consente di avere dei feedback iniziali da parte del cliente. Questi rilasci frequenti consentiranno di avere un ritorno anticipato sull'investimento.

2.2. La Servant Leadership

La servant leadership è uno stile di leadership basato sull'idea che il ruolo del leader sia quello di servire gli altri, in particolare i membri del team, prima di sé stessi. I leader servant cercano di promuovere un ambiente di lavoro inclusivo e collaborativo, incoraggiando i membri del team a sviluppare le loro competenze e a prendere decisioni. Si concentrano sul supporto e sulla crescita dei team piuttosto che sulla gerarchia e il potere, e cercano di creare un ambiente di lavoro in cui ognuno possa sentirsi valorizzato e soddisfatto. La servant leadership è spesso associata a un aumento della soddisfazione del lavoro, della motivazione e della collaborazione all'interno del team.

È stata fatta la scelta di seguire questo metodo in modo da responsabilizzare il team effettuando azioni di incoraggiamento e supporto. Così facendo, i team member potranno sviluppare nuove skill e acquisire fiducia nelle proprie capacità.

2.3. Scrum

Scrum è un framework di sviluppo agile utilizzato per gestire progetti di software. Si basa sulla filosofia Agile, che sostiene la collaborazione e il cambiamento continuo per la creazione di prodotti di alta qualità. In Scrum, i progetti sono suddivisi in piccole iterazioni chiamate "sprint", in cui un team di sviluppo lavora insieme per completare una serie di compiti. Al termine di ogni sprint, il team riunisce per una riunione di revisione (demo) in cui presenta il lavoro completato e riceve il feedback dal product owner (il proprietario del prodotto).



Laurea Magistrale in informatica-Università di Salerno Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

Scrum utilizza una serie di ruoli specifici, come il product owner, il team di sviluppo e lo Scrum Master (che funge da facilitatore del processo). Inoltre, utilizza una serie di artefatti, come il backlog del prodotto (una lista di compiti da completare) e il diagramma di burndown (che traccia il progresso del progetto).

Nel processo Scrum, il product owner definisce gli obiettivi del progetto e stabilisce la priorità dei compiti nel backlog del prodotto. Il team di sviluppo, composto da ingegneri esperti, lavora insieme per completare i compiti del backlog durante uno sprint. Lo Scrum Master funge da facilitatore, assicurandosi che il team lavori in modo efficiente e senza ostacoli. Al termine di ogni sprint, il team tiene una riunione di revisione (demo) in cui presenta il lavoro completato e riceve il feedback dal product owner. Il diagramma di burndown viene utilizzato per tracciare il progresso del progetto e identificare eventuali problemi o ritardi.

Scrum è una metodologia di sviluppo agile popolare perché è flessibile e incentrata sulla collaborazione, il che consente di adattarsi rapidamente ai cambiamenti e alle esigenze dei clienti. Inoltre, incoraggia la trasparenza e il feedback, il che aiuta a identificare e risolvere rapidamente i problemi durante il processo di sviluppo.

2.4. Kanban

Kanban è una metodologia di gestione dei progetti basata sulla visualizzazione del flusso di lavoro e sulla limitazione della quantità di lavoro in corso. La parola "Kanban" è giapponese e significa "cartello". In Kanban, il flusso di lavoro viene visualizzato su una scheda o una bacheca, con ogni compito rappresentato da una scheda o una carta.

La metodologia Kanban viene spesso utilizzata in combinazione con la metodologia Agile, ma può essere utilizzata anche in altri contesti. In Kanban, il team lavora su una serie di compiti, che vengono spostati attraverso diverse colonne sulla bacheca man mano che vengono completati. Ogni colonna rappresenta una fase del processo, ad esempio "da fare", "in corso" e "completato".

Il concetto chiave di Kanban è la limitazione del lavoro in corso (WIP, work in progress). Ciò significa che il team si impegna a lavorare su un numero limitato di compiti alla volta, in modo da evitare il sovraccarico e il burnout. Ciò consente al team di concentrarsi su una quantità gestibile di lavoro, aumentando l'efficienza e riducendo i tempi di consegna.

Kanban è una metodologia flessibile che può essere adattata a diverse situazioni e contesti. Offre una visibilità costante del flusso di lavoro e consente al team di identificare rapidamente i problemi e intervenire per risolverli. Inoltre, incoraggia la collaborazione e il feedback, il che aiuta a mantenere il team allineato e a soddisfare le esigenze dei clienti.

2.5. Extreme Programming

Extreme Programming (XP) è una metodologia di sviluppo agile che si concentra sulla collaborazione, la comunicazione e la qualità del codice. XP sostiene che il lavoro di un team di sviluppo può essere reso più efficace e di qualità superiore attraverso la pratica di alcune "regole estreme" o principi.



Laurea Magistrale in informatica-Università di Salerno Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

I principi di XP includono il lavoro in coppia, in cui due sviluppatori lavorano insieme su un singolo compito; il testing continuo, in cui il team esegue test di qualità del codice durante tutto il processo di sviluppo; e il rilascio continuo, in cui il team rilascia regolarmente piccole parti del software in modo da poter ricevere feedback costante e apportare modifiche in tempo reale.

XP sottolinea anche l'importanza della comunicazione e della collaborazione, nonché della flessibilità e dell'adattamento ai cambiamenti. Si ritiene che questi principi aiutino a creare software di alta qualità in modo più efficiente e soddisfacente per il team.

2.6. Mix di approcci

Per la fase di implementazione verrà utilizzato l'approccio Agile "Scrum" con qualche elemento di Kanban ed Extreme Programming (XP).

In sintesi, verranno utilizzate le linee guida sull'utilizzo degli Sprint, delle pianificazioni dello sprint, il Daily Scrum Meeting, Sprint Review e sessioni di retrospective di sprint dal metodo Scrum.

Una Kanban Board per aiutare il team ad organizzare il lavoro, attraverso l'utilizzo di Trello. Ciò aiuterà ad avere una visione chiara degli impedimenti e del lavoro da eseguire.

La tecnica di Pair programming verrà utilizzata nel progetto EnIA. Tale pratica è alla base del modello di Extreme Programming.

3. Motivazione scelte

È risaputo che attraverso di tecniche e approcci agili è possibile gestire al meglio lo sviluppo di un progetto software. Successivamente alla fase di design che è stata svolta utilizzando un modello tradizionale, ci si è accorti che l'utilizzo dei modelli Agile poteva essere una soluzione di sviluppo migliore. Perciò, durante la fase di implementazione, si è deciso di convergere nei modelli Agile, sfruttando come modello principale Scrum, ma con delle sfumature di Kanban e Extreme Programming.