

Portale di un'istituzione educativa

Progetto di Ingegneria del Software 2021/2022

Professore Davide Rossi

Tutor Fiorela Ciroku

Descrizione del metodo adottato

Gruppo di lavoro:

- Magazzù Gabriele - 0000933823.
- Marchese Renato Junior - 0000949773.
- Monti Stefano - 0000948005.

Documentazione relativa alla descrizione del metodo adottato per la gestione e sviluppo del progetto.

Sommario

[Descrizione del metodo adottato](#)

[Metodologia Scrum](#)

[Ruoli](#)

[Stakeholders](#)

[Product Owner](#)

[Scrum Master](#)

[Development team](#)

[Artefatti](#)

[Product Backlog](#)

[Sprint Backlog](#)

[Burn Down Chart](#)

[Eventi](#)

[Daily Scrum](#)

[Sprint Planning](#)

[Sprint Review & Sprint Retrospective](#)

[Milestone](#)

[User Stories](#)

[Conclusioni](#)

Metodologia Scrum

Il Framework **Scrum** fa parte delle metodologie di sviluppo **Agile** e comprende tutte le pratiche rivolte alla gestione organizzativa del progetto che si vuole realizzare.

In quanto metodologia Agile il framework presenta tutte le loro caratteristiche principali, specializzandosi su alcune di esse per favorire una migliore gestione organizzativa del progetto.

L'idea principale è di rimanere in stretto contatto con gli **stakeholders** (committenti del progetto), o da chi ne fa le veci, per poter concretizzare al meglio le richieste implementative.

Il Framework divide i progressi in fasi (o iterazioni) chiamate **Sprint**, ognuna con una deadline (iterazioni time-boxed) ed ognuna include una progettazione completa, sviluppo del codice, cicli di test, e termina con un potenziale incremento del **valore** del progetto.

Il Framework è composto da ruoli, artefatti ed eventi.

Il gruppo ha suddiviso il lavoro in quattro fasi principali:

1. **Pre-Inception**: fase iniziale volta alla lettura e comprensione delle specifiche progettuali, all'installazione delle tecnologie necessarie e alla creazione / preparazione degli strumenti necessari.
2. **Inception**: fase intermedia volta alla definizione degli obiettivi e requisiti da implementare (**milestone**), requisiti funzionali all'utente (**user stories**) e altri requisiti non necessari per il funzionamento, ma importanti per il progetto.
Per definire gli obiettivi e i requisiti vengono studiati e realizzati sempre in questa fase il diagramma di dominio e il diagramma dei casi d'uso relativi al progetto, la cui versione finale la si può trovare nella seguente cartella: [diagrammi casi d'uso e dominio](#).
3. **Construction**: fase di sviluppo e di "costruzione" del progetto tramite la gestione degli eventi tipici di Scrum, in particolare gli Sprint.
Sono stati gestiti l'architettura client-server tipica di GWT; è stata gestita l'inizializzazione e la costruzione delle collezioni necessarie in un database MapDB per salvare i dati; è stata scelta e costruita la grafica / interfaccia delle pagine del portale.
La fase di construction ha compreso anche una fase di debugging tramite prove manuali e unit testing tramite metodi appositi con JUnit.
É consultabile una relazione sulle scelte progettuali e di interfaccia al seguente link: [scelte progettuali e di interfaccia](#).
Il manuale per installare e lanciare il progetto è invece consultabile al seguente link: [Manuale](#).
4. **Revisioning**: fase di revisione nella quale si effettua un ulteriore debug e test del codice, e nella quale si è svolta la stesura della documentazione.

Ruoli

Il framework è caratterizzato dalla presenza di vari ruoli, fondamentali per la gestione di un progetto tramite la metodologia Agile Scrum.

I ruoli principali sono: **stakeholders**, **product owner**, **scrum master** e **development team**.
Esistono ruoli addizionali, come costumer ed executive manager.

Nel caso del progetto a noi commissionato potremmo identificare gli stakeholders nel Professore Rossi e nella Tutor Ciroku.

I membri del development team sono identificati dai membri del gruppo di lavoro.

Product owner e scrum master sono identificati da due membri del team, diversi per ogni sprint (solitamente sono soggetti esterni al team di sviluppo).

Stakeholders

Gli stakeholders sono i committenti del progetto che deve essere realizzato. Vengono rappresentati all'interno dell'azienda che si occupa della realizzazione del progetto dal product owner.

Product Owner

Rappresenta gli stakeholder all'interno dell'azienda che progetta e sviluppa il software. Definisce i requisiti, le user stories e il loro livello di priorità (low, medium, high).

Scrum Master

Identifica il leader in servizio nel team. Presenta le competenze specifiche di tipo software e le mette a disposizione del team nel ruolo di leader e di facilitatore nel caso si presentassero ostacoli.

Development team

Gruppo di persone (solitamente da 3 a 9) che sviluppa il progetto, realizzando le task associate ad ogni sprint. Il team è autonomo e si assegnano internamente i ruoli per lo sviluppo del sistema.

Artefatti

Il framework Scrum fa utilizzo di alcuni elementi, chiamati artefatti, per la gestione di un progetto. I principali artefatti sono i seguenti:

- **Product Backlog.**
- **Sprint Backlog.**
- **Burn Down Chart.**

Il gruppo ha fatto uso dello strumento **Trello** (messo a disposizione dallo spazio di lavoro Atlassian) come Product Backlog e ha fatto uso dello strumento **Jira** (Issue Tracker gratuito di Atlassian) come Sprint Backlog. Inoltre per ogni Sprint è stato generato un grafico Burn Down Chart (attraverso Jira) ed è stato salvato in formato PNG.

Product Backlog

Il Product Backlog consiste in una lista ordinata delle cose da fare. La lista è redatta dal Product Owner, in base alle priorità, e comprende: **milestone, user stories, bug fixes, requisiti non funzionali, requisiti tecnologici** e **chores** (oggetti che producono valore al team, ma non direttamente agli stakeholders).
[Trello](#).

Sprint Backlog

Lo Sprint Backlog consiste in una lista ordinata delle **task** da realizzare all'interno di uno specifico Sprint, quindi entro una **deadline**.

La lista è concordata dal team di sviluppo insieme al Product Owner, convertendo ogni milestone, user storie o requisito da implementare in una o più task associate, e calcolandone l'**effort**, ovvero una stima dell'ammontare di lavoro da dedicarci.

[Roadmap Jira.](#)

Burn Down Chart

Il Burn Down Chart è un diagramma che descrive la variazione della realizzazione delle task nel tempo e la variazione dell'effort associato.

[Diagrammi burn down chart.](#)

Eventi

Il framework Scrum è caratterizzato da alcuni momenti di confronto tra i membri del team e i committenti (o chi ne fa le veci). I principali eventi sono i seguenti:

- **Daily Scrum.**
- **Sprint Planning.**
- **Sprint Review.**
- **Sprint Retrospective.**

Il gruppo ha fatto uso di tutte le tipologie di eventi.

Daily Scrum

Il Daily Scrum consiste in una riunione giornaliera, informale. Di norma viene effettuata in piedi per un tempo che solitamente non supera i 15 minuti.

E' organizzata e tenuta dallo Scrum Master e si risponde alle seguenti tre domande:

1. Cosa è stato fatto ieri?
2. Cosa si farà oggi?
3. Ci sono stati problemi nel raggiungimento di un particolare obiettivo?

Il gruppo, data la situazione di emergenza sanitaria mondiale dovuta alla pandemia, ha svolto i Daily Scrum in modalità telematica tramite l'utilizzo di piattaforme di comunicazione quali **Microsoft Teams**, **Discord** e **WhatsApp**.

Il gruppo ha inoltre stilato un **diario** schematico per tenere traccia di cosa è stato svolto all'interno di ogni singola giornata di lavoro.

E' stato effettuato un Daily Scrum ogni qual volta il team si riuniva per lavorare insieme.

Nel diario non si è tenuta traccia delle modifiche apportate da lavori individuali svolti singolarmente dai membri del team. Nonostante ciò, il gruppo è sempre rimasto aggiornato di ogni modifica effettuata, appositamente comunicate tramite una delle piattaforme di comunicazione di cui sopra.

[Diario.](#)

Sprint Planning

Lo Sprint Planning è un evento fondamentale nel quale si pianifica **come popolare** lo Sprint Backlog.

Consta di due fasi:

1. Popolazione del Product Backlog da parte del Product Owner, con relativa stima dell'effort e discussione con il team di sviluppo.
2. Popolazione dello Sprint Backlog da parte del team di sviluppo, che converte le milestone, le user stories e tutti i requisiti implementativi in una o più task associate, ognuna con il relativo effort.

L'effort può essere stimato tramite:

- **velocity-driven planning** → stima dell'effort in termini di tempo necessario per realizzare le task tramite assegnamento degli **story points**.
- **capacity-driven planning** → stima dell'effort in termini di tempo necessario per realizzare le task tramite la valutazione delle capacità di ogni singolo membro del team.

Il gruppo ha utilizzato una via intermedia per la stima dell'effort.

In primo luogo è stata stimata la difficoltà delle task tramite assegnamento di story points, riprendendo lo schema della **sequenza di Fibonacci**, quindi in maniera esponenziale.

In secondo luogo sono state valutate le capacità di ogni singolo membro del team e sono state rivalutate alcune task, modificando gli effettivi story points associati.

Il gruppo ha effettuato un **primo** Sprint Planning e ha successivamente fatto corrispondere tale evento con lo Sprint Retrospective.

Il gruppo, a causa della poca esperienza e di altre necessità personali, ha scelto di adottare una durata settimanale o bisettimanale per ogni Sprint.

Alcuni sprint hanno avuto la necessità di una durata maggiore per poter portare a termine le task inserite nel backlog.

In totale sono stati completati **6** Sprint.

[Primo Sprint Planning](#).

Sprint Review & Sprint Retrospective

Lo Sprint Review e lo Sprint Retrospective sono eventi tipici del framework Scrum.

Lo Sprint Review si occupa della revisione di quanto è stato fatto nello Sprint appena terminato: si inquadrano eventuali ostacoli, difficoltà riscontrate e si fa una valutazione di quanto è stato portato a termine e quanto è rimasto da completare.

Lo Sprint Retrospective si occupa della discussione delle specifiche del prossimo Sprint da iniziare.

Il gruppo ha tenuto traccia delle riunioni di Sprint Review e Sprint Retrospective in un apposito file.

[Sprint Review & Retrospective](#).

Milestone

Le milestone corrispondono agli obiettivi fondamentali da raggiungere e da implementare nel progetto.

Sono state individuate le seguenti milestone:

- Completamento del progetto.
- Installazione strumenti necessari.
- Realizzazione del diagramma di dominio.

- Realizzazione del diagramma dei casi d'uso.
- Creazione di tutte le classi Java del dominio del sistema.
- Realizzazione homepage.
- Registrazione al sito (Form Sign In).
- Accesso al sito (Form Log In).
- Meccanismo sessioni per gestione login-logout.
- User interface appropriata per ogni tipologia di utente.
- Meccanismo di inserimento dei dati nel Database.
- Meccanismo di modifica dei dati nel Database.
- Visualizzazione dati (informazioni personali, voti, etc.).
- Gestione dell' invio dei voti (docente → segreteria).
- Gestione della ricezione e pubblicazione dei voti (segreteria → portale).
- Ciclo di test completo del sito.

User Stories

Le user stories rappresentano le specifiche del progetto, scritte nel linguaggio di dominio, che catturano le aspettative e le interazioni degli utenti con il sistema.

Una user story ha la seguente struttura:

As a **<type of user>**, I want **<some goal>** so that **<some reason>**.

Una user story deve essere indipendente rispetto alle altre, e bisogna stimarne un effort.

Le user stories che abbiamo dedotto dalle specifiche del progetto sono:

- **In quanto** Utente **voglio** accedere e registrarmi al portale **per** interagire con l'istituzione educativa.
- **In quanto** Studente **voglio** poter visualizzare i corsi disponibili **per** potermi iscrivere.
- **In quanto** Studente **voglio** poter effettuare l'iscrizione **per** frequentare il corso scelto.
- **In quanto** Studente **voglio** poter visualizzare le mie informazioni personali **per** controllare che siano corrette.
- **In quanto** Studente **voglio** poter effettuare la registrazione ad un esame **per** poterlo svolgere.
- **In quanto** Studente **voglio** poter visualizzare i voti degli esami da me svolti **per** conoscere il loro esito.
- **In quanto** Docente **voglio** poter creare un corso **per** permettere agli studenti di seguirlo.
- **In quanto** Docente **voglio** poter cancellare un corso **per** rimuoverlo dai corsi disponibili.
- **In quanto** Docente **voglio** poter modificare un corso **per** correggere o aggiungere eventuali informazioni relative al corso stesso.
- **In quanto** Docente **voglio** poter visualizzare le mie informazioni personali **per** controllare che siano corrette.
- **In quanto** Docente **voglio** poter aggiungere un esame **per** permettere agli studenti di iscriversi e sostenerlo.
- **In quanto** Docente **voglio** poter modificare un esame **per** correggere eventuali errori.
- **In quanto** Docente **voglio** poter cancellare un esame **per** rimuoverlo dalla lista degli esami disponibili.

- **In quanto** Docente **voglio** poter inviare i voti degli esami svolti alla segreteria **per** poterli pubblicare e verbalizzare.
- **In quanto** Segreteria **voglio** poter visualizzare le informazioni degli studenti **per** verificare che siano complete.
- **In quanto** Segreteria **voglio** poter pubblicare i voti degli esami **per** renderli visibili agli studenti.
- **In quanto** Segreteria **voglio** poter verbalizzare i voti degli esami **per** certificare il risultato dell'esame.
- **In quanto** Admin **voglio** poter creare gli account degli utenti di tipo studente, docente e segreteria **per** permettere loro di accedere al portale.
- **In quanto** Admin **voglio** poter aggiungere le informazioni relative agli utenti di tipo studente e docente **per** permettere loro di visualizzarle.
- **In quanto** Admin **voglio** poter modificare le informazioni relative agli utenti di tipo studente e docente **per** correggere eventuali errori o aggiungere sezioni mancanti.
- **In quanto** Admin **voglio** poter visualizzare le informazioni relative agli utenti di tipo studente e docente **per** verificare che siano corrette.

Oltre alle precedenti milestone e user stories sono poi state aggiunte ulteriori task derivanti da necessità riscontrate in fase di sviluppo del progetto.

Tutte le task eseguite possono essere trovate sul file riassuntivo degli sprint ([Sprint](#)), che riassume gli sprint inseriti e completati su Jira.

Conclusioni

Il progetto è stato portato a termine con molte difficoltà riscontrate: tali vanno dall'ambito più puramente informatico all'ambito più personale dei singoli membri del team.

Il team si ritiene comunque contento, nonostante i molti errori commessi durante la fase di Construction, di come ha lavorato: per l'impegno dimostrato e la collaborazione nelle varie task.

Infatti, nonostante il lavoro sia stato diviso in modo più o meno equo, spesso si lavorava utilizzando una tecnica di peer coding review: tramite la condivisione dello schermo un membro del team lavorava e gli altri controllavano che tutto seguisse la logica della modellazione del progetto e le regole di buona programmazione che avevamo precedentemente stabilito di seguire.