

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA INFORMATICA ED ELETTRICA E MATEMATICA APPLICATA



Laurea Magistrale in Ingegneria Informatica

Relazione Project Work - Software Architecture & Design - ShapeTastic

Gruppo N°16 - AH

Studenti:

Cutolo Ciro	0622702532	c.cutolo7@studenti.unisa.it
Frasca Gaetano	0622702610	g.frasca10@studenti.unisa.it
D'Ambrosio Gennaro	0622702464	g.dambrosio65@studenti.unisa.it
Bottiglieri Alessio	0622702583	a.bottiglieri16@studenti.unisa.it

Sommario

1	Pre-game	4
1.1	Initial product backlog	4
1.1.1	Basic operation of the program	4
1.1.2	Shape editing	7
1.1.3	Display of the drawing	10
1.1.4	More shapes and editing operations	13
1.1.5	Groups and shape libraries	15
1.2	Definition of Done	17
1.3	Description of software architecture	18
1.3.1	Architettura	18
1.4	Architecture and technologies chosen	19
1.5	Mock-up interface	19
2	First Sprint	20
2.1	First sprint Planning	20
2.1.1	Stima iniziale della Project Velocity	20
2.1.2	First Sprint Backlog	20
2.1.3	Suddivisione in task e assegnazione	21
2.2	First Sprint - Applicazione Pattern	26
2.2.1	Simple Factory Pattern	26
2.2.2	Prototype Pattern	27
2.3	First Sprint - Testing	28
2.4	First Sprint Backlog	28
2.5	Updated Product Backlog	30
2.6	Project Burndown chart:	31
2.7	First Sprint - Review	32
2.7.1	Project velocity measured:	32
2.7.2	Problemi riscontrati durante l'implementazione	32
2.7.3	Modifiche da apportare al Product Backlog	32
2.8	First Sprint - Retrospective	33
2.8.1	Starfish Diagram	33
3	Second sprint	34
3.1	Second Sprint Planning	34
3.1.1	Modifiche Product Backlog	34
3.1.2	Stima aggiornata della Project Velocity	35

3.1.3	Selezione delle User Stories per la seconda sprint	35
3.1.4	Suddivisione in task e assegnazione	36
3.2	Second Sprint - Applicazione Pattern	40
3.2.1	Command Pattern	40
3.3	Second Sprint Backlog.....	41
3.4	Updated Product Backlog	42
3.5	Project Burndown chart:	43
3.6	Second Sprint - Review	44
3.6.1	Project velocity measured	44
3.6.2	Problemi riscontrati durante l'implementazione	44
3.6.3	Modifiche da apportare al Product Backlog	44
3.7	Second Sprint - Retrospective.....	45
3.7.1	Starfish Diagram	45
4	Third sprint	46
4.1	Third Sprint Planning.....	46
4.1.1	Modifiche Product Backlog.....	46
4.1.2	Stima aggiornata della Project Velocity.....	46
4.1.3	Selezione delle User Stories per la seconda sprint	47
4.1.4	Suddivisione in task e assegnazione	47
4.2	Third Sprint - Applicazione Pattern	51
4.2.1	Command Pattern: Correzione.....	51
4.3	Third Sprint Backlog.....	52
4.4	Updated Product Backlog	53
4.5	Project Burndown chart:	55
4.6	Third Sprint - Review	56
4.6.1	Project velocity measured:	56
4.6.2	Problemi riscontrati durante l'implementazione	56
4.6.3	Modifiche da apportare al Product Backlog	56
4.7	Third Sprint - Retrospective.....	57
4.7.1	Starfish Diagram	57

1 Pre-game

Link utili:

- Trello Board: <https://trello.com/b/1OiONvWL/pw16-sad>
- Repository GitHub: https://github.com/CiroCutolo/Group16_SoftwareArchitecture-Design_ProjectWork

1.1 Initial product backlog

1.1.1 Basic operation of the program

1.1.1.1 Finestra iniziale

Story:

Come utente,
Vorrei vedere una finestra inizialmente vuota,
Così che possa iniziare a disegnare su di essa.

Acceptance Criteria:

- Dato che l'applicazione è stata appena avviata,
Quando l'interfaccia utente viene caricata,
Allora l'utente vede una finestra inizialmente vuota.

Priority: Alta

Story Points: 1

1.1.1.2 Forme supportate

Story:

Come utente,
Vorrei poter scegliere tra segmento di linea, rettangolo ed ellisse,
Così che possa creare disegni utilizzando queste forme.

Acceptance Criteria:

- Dato che l'interfaccia è visibile,
Quando l'utente guarda la sezione con l'elenco delle forme,
Allora vede tre opzioni: linea, rettangolo ed ellisse,

Priority: Alta

Story Points: 3

1.1.1.3 Aggiunta forme geometriche

Story:

Come utente,
Vorrei poter selezionare una forma geometrica tra le disponibili e posizionarla con il mouse,
Così che possa aggiungere quella forma in un punto specifico della finestra.

Acceptance Criteria:

- Dato che l'utente non ha selezionato alcuna forma,
Quando effettua un clic sull'area di disegno,

Allora non deve essere eseguita alcuna azione e non deve comparire nessuna anteprima.

- Dato che l'utente ha selezionato una forma tra quelle disponibili,
Quando preme il tasto sinistro del mouse sull'area di disegno,
Allora deve iniziare la fase di disegno, memorizzando le coordinate iniziali del puntatore.
- Dato che l'utente sta trascinando il mouse dopo aver cliccato sull'area di disegno,
Quando sposta il cursore tenendo premuto il tasto,
Allora deve essere mostrata in tempo reale un'anteprima visiva della forma, aggiornata dinamicamente in base alla posizione del cursore.
- Dato che l'utente ha rilasciato il mouse e la forma è stata inserita,
Quando termina l'interazione di disegno,
Allora l'anteprima temporanea deve scomparire, lasciando visibile solo la forma definitiva.
- Dato che l'utente ha completato il trascinamento,
Quando rilascia il tasto sinistro del mouse,
Allora la forma selezionata deve essere inserita in modo definitivo nell'area di disegno,
con le dimensioni e la posizione corrispondenti all'azione eseguita.

Priority: Alta

Story Points: 3

1.1.1.4 Scelta colore del bordo delle forme

Story:

Come utente,
Vorrei poter scegliere il colore del bordo della forma da aggiungere,
Così che possa personalizzarne l'aspetto.

Acceptance Criteria:

- Dato che l'interfaccia è visibile, quando l'utente guarda la sezione "Colore bordo", allora vede un selettore contenente almeno 8 colori
- Dato che una forma non è ancora stata disegnata, quando l'utente seleziona un colore per il bordo, allora il colore selezionato viene associato alla forma da disegnare
- Dato che l'utente ha selezionato un colore per il bordo, quando disegna una nuova forma, allora la forma presenta un bordo del colore scelto

Priority: Alta

Story Points: 3

1.1.1.5 Scelta colore di riempimento delle forme

Story:

Come utente,
Vorrei scegliere il colore di riempimento delle forme chiuse (come l'ellissi),
Così che possa personalizzarne l'aspetto.

Acceptance Criteria:

- Dato che l'interfaccia è visibile, quando l'utente guarda la sezione "Colore Riempimento", allora vede un selettore contenente almeno 8 colori
- Dato che l'interfaccia è visibile, quando seleziona un colore di riempimento nell'apposita sezione, allora il colore viene associato alla prossima forma chiusa da disegnare
- Dato che un colore di riempimento è stato selezionato, quando l'utente disegna una forma chiusa, allora l'interno della forma viene riempito con il colore scelto

Priority: Alta

Story Points: 3

1.1.1.6 Salvataggio dei disegni

Story:

Come utente,
Vorrei poter salvare il disegno in un file,
Così che possa conservarlo sul mio dispositivo.

Acceptance Criteria:

- Dato che l'interfaccia è visibile, quando l'utente guarda l'interfaccia, allora vede un bottone "Salva"
- Dato che il bottone "Salva" è visibile, quando l'utente lo clicca, allora si apre una finestra di dialogo per scegliere dove salvare il file
- Dato che l'utente ha scelto un percorso di salvataggio, quando conferma l'azione, allora il disegno viene salvato correttamente in un formato immagine

Priority: Alta

Story Points: 5

1.1.1.7 Caricamento dei disegni

Story:

Come utente,
Vorrei poter caricare un disegno precedentemente salvato,
Così che possa continuare a lavorarci.

Acceptance Criteria:

- Dato che l'interfaccia è visibile, quando l'utente guarda l'interfaccia, allora è presente un bottone "Carica"
- Dato che il bottone "Carica" è visibile, quando l'utente lo clicca, allora si apre una finestra di selezione file per scegliere un disegno nel formato immagine supportato
- Dato che l'utente ha selezionato un file, quando conferma la scelta, allora il disegno viene caricato correttamente nell'area di disegno dell'applicazione

Priority: Alta

Story Points: 5

1.1.2 Shape editing

1.1.2.1 Selezione forma

Story:

Come utente,
Voglio poter selezionare una forma già disegnata e applicarvi un'operazione tramite mouse,
in modo da poterne variare le proprietà.

Acceptance criteria:

- Data una forma presente nel riquadro di disegno,
Quando clicco la forma con il mouse,
Allora questa deve risultare selezionata e pronta per essere manipolata.

Priority: Medio - Alta

Story points: 3

1.1.2.2 Eliminazione di una forma

Story:

Come utente,
Voglio poter eliminare la forma selezionata,
in modo da poterla cancellare liberamente dal riquadro di disegno.

Acceptance criteria:

- Data una forma selezionata,
Quando clicco l'opzione "elimina",
Allora la forma deve essere rimossa dal riquadro di disegno.

Priority: Medio - Alta

Story points: 1

1.1.2.3 Copia e Incolla

Story:

Come utente,
Voglio poter copiare e incollare le forme selezionate,
in modo da poterle salvare negli appunti e riutilizzare nel riquadro di disegno.

Acceptance criteria:

- Date delle forme selezionate,
Quando clicco sull'opzione "copia",
Allora la selezione deve essere salvata negli appunti.
- Date delle forme salvate negli appunti,
Quando clicco l'opzione "incolla",
Allora le forme devono comparire nel riquadro nella posizione del cursore al momento del click.

Priority: Medio - Alta

Story points: 3

1.1.2.4 Taglia

Story:

Come utente,
voglio poter tagliare le forme selezionate,
in modo da rimuoverle dal riquadro di disegno e copiarle negli appunti.

Acceptance criteria:

- Date una o più forme selezionate,
Quando clicco sull'opzione "taglia",
Allora le forme devono essere rimosse dal riquadro di disegno e copiate negli appunti.

Priority: Medio - Alta

Story points: 2

1.1.2.5 Undo

Story:

Come utente,
voglio poter annullare illimitatamente le operazioni precedentemente effettuate,
in modo da poter tornare a versioni precedenti del riquadro di disegno.

Acceptance criteria:

- Date delle operazioni effettuate,
Quando clicco il tasto apposito per effettuare l'annullamento (undo),
Allora deve essere annullata l'ultima azione performata nel riquadro di disegno.

Priority: Medio - Alta

Story points: 5

1.1.2.6 Modifica della posizione di una forma

Story:

Come utente,
voglio poter modificare la posizione della forma selezionata,
in modo da poterla spostare liberamente all'interno del riquadro di disegno.

Acceptance criteria:

- Data una forma selezionata,
Quando la trascino con il mouse,
Allora la forma deve seguire il movimento del cursore e aggiornare la sua posizione nel riquadro in tempo reale.

Priority: Medio - Alta

Story points: 2

1.1.2.7 Modifica del colore di una forma

Story:

Come utente,
voglio poter modificare il colore, sia del contorno che dell'interno, della forma selezionata,
in modo da personalizzarne liberamente l'aspetto.

Acceptance criteria:

- Data una forma selezionata,
Quando modifico il colore tramite la sezione apposita,
Allora la forma deve aggiornare il suo aspetto in tempo reale, riflettendo i nuovi colori scelti.

Priority: Medio - Alta

Story points: 2

1.1.2.8 Modifica delle dimensioni di una forma

Story:

Come utente,
voglio poter modificare le dimensioni della forma selezionata,
in modo da ingrandirla o ridurla liberamente all'interno del riquadro di disegno.

Acceptance criteria:

- Data una forma selezionata,
Quando modifico le dimensioni tramite menu contestuale,
Allora la forma deve cambiare la propria dimensione.

Priority: Medio - Alta

Story points: 5

1.1.2.9 Modifica del livello di una forma

Story:

Come utente,
voglio poter modificare il livello di una forma o di un gruppo di forme,
in modo da controllare la sovrapposizione e portarle in piani differenti.

Acceptance criteria:

- Dato una o più forme presenti nel riquadro di disegno,
Quando seleziono "Porta in primo piano",
Allora la forma, o il gruppo di forme, deve/devono comparire sopra tutte le altre.
- Dato una o più forme presenti nel riquadro di disegno,
Quando seleziono il comando apposito (ad es. "Porta avanti di un livello"),
Allora la forma, o il gruppo di forme, deve/devono spostarsi di un livello sopra rispetto all'attuale posizione.
- Dato una o più forme presenti nel riquadro di disegno,
Quando seleziono il comando apposito (ad es. "Porta indietro di un livello"),
Allora la forma, o il gruppo di forme, deve/devono spostarsi di un livello sotto rispetto all'attuale posizione.

- Dato una o più forme presenti nel riquadro di disegno,
Quando seleziono "Manda in ultimo piano",
Allora la forma, o il gruppo di forme, deve/devono comparire sotto tutte le altre.

Priority: Medio - Alta

Story points: 3

1.1.3 Display of the drawing

1.1.3.1 *Modifica del livello di zoom*

Story:

Come utente,
Voglio poter modificare il livello di zoom dell'area di disegno,
Così da poter osservare l'intero disegno o i suoi dettagli.

Acceptance criteria:

- Dato un disegno visualizzato nella canvas,
Quando l'utente seleziona un nuovo livello di zoom tramite bottoni, slider o menu,
Allora il livello di zoom viene aggiornato correttamente tra almeno quattro valori predefiniti (es. 25%, 50%, 100%, 200%).
- Dato un qualsiasi livello di zoom attivo,
Quando l'utente modifica lo zoom,
Allora l'interfaccia mostra chiaramente il nuovo livello di zoom selezionato.
- Dato un disegno composto da una o più forme,
Quando l'utente effettua uno zoom in o out,
Allora le forme mantengono proporzioni corrette e posizione coerente all'interno dell'area di disegno.

Priority: Media

Story points: 5

1.1.3.2 *Area di disegno espandibile*

Story:

Come utente,
voglio che l'area di disegno possa essere più grande della finestra,
così da poter creare disegni senza essere limitato dallo spazio visibile.

Acceptance criteria:

- Data un'area di disegno attiva,
Quando l'utente crea un disegno più grande della finestra,
Allora l'area si espande dinamicamente per contenere tutto il contenuto, senza perdere o tagliare alcuna parte.
- Dato un disegno che si estende oltre i limiti visibili della finestra,
Quando l'utente inserisce o modifica forme in una zona non visibile,

Allora tali forme vengono correttamente gestite, visualizzate e conservate anche se si trovano al di fuori dell'area attualmente visibile.

Priority: Media

Story points: 3

1.1.3.3 Scroll area di disegno

Story:

Come utente,
voglio poter scorrere l'area di disegno,
così da visualizzare e modificare qualsiasi parte del mio disegno in caso l'area di disegno sia più grande dell'interfaccia.

Acceptance Criteria:

- Data un'area di disegno con dimensioni maggiori rispetto alla finestra,
Quando l'utente visualizza l'area,
Allora vengono mostrate automaticamente una scrollbar orizzontale e una verticale per permettere lo scorrimento.
- Data un'area di disegno con scrollbar visibili,
Quando l'utente le utilizza per navigare,
Allora può visualizzare correttamente ogni parte del disegno, senza perdita di contenuto né distorsioni.

Priority: Media

Story points: 2

1.1.3.4 Visualizzazione della griglia

Story:

Come utente,
voglio poter visualizzare una griglia sull'area di disegno,
così da posizionare più facilmente le forme.

Acceptance criteria:

- Data un'area di disegno con forme già inserite,
Quando l'utente attiva la visualizzazione della griglia,
Allora vengono mostrate linee orizzontali e verticali equidistanti, posizionate sopra lo sfondo ma sotto le forme.
- Data una griglia visibile,
Quando l'utente interagisce con le forme (es. selezione o trascinamento),
Allora le linee della griglia non interferiscono con tali operazioni.
- Data la visualizzazione della griglia attivata,
Quando l'area di disegno viene aggiornata (es. tramite scroll o zoom),
Allora la griglia si muove coerentemente con la canvas e adatta la distanza tra le linee proporzionalmente al livello di zoom.

Priority: Media

Story points: 3

1.1.3.5 Attivazione e disattivazione della griglia

Story:

Come utente,
voglio poter attivare o disattivare la griglia,
così da poterla usare solo se necessario.

Acceptance criteria:

- Data una sezione dell'interfaccia con un controllo per la griglia (es. checkbox o voce di menu),
Quando l'utente attiva o disattiva l'opzione,
Allora la griglia viene mostrata o nascosta in tempo reale nell'area di disegno.
- Dato lo stato corrente della griglia (attiva o disattiva),
Quando l'utente utilizza altre funzionalità (es. inserimento di forme, zoom),
Allora lo stato della griglia rimane invariato e il suo stato è chiaramente visibile nell'interfaccia.

Priority: Media

Story points: 1

1.1.3.6 Cambiare dimensioni griglia

Story:

Come utente,
voglio poter scegliere la dimensione della griglia,
così da poterla adattare meglio al tipo di disegno che sto facendo.

Acceptance Criteria:

- Data una griglia visibile sull'area di disegno,
Quando l'utente seleziona una nuova dimensione tra i valori predefiniti (es. 10, 20, 40 unità logiche),
Allora la griglia si aggiorna in tempo reale, mostrando linee equidistanti e proporzionate anche in presenza di zoom.
- Data una griglia disattivata,
Quando l'utente modifica la dimensione della griglia,
Allora non viene mostrata alcuna modifica visiva, ma il nuovo valore viene memorizzato e applicato alla successiva attivazione.

Priority: Media

Story points: 3

1.1.4 More shapes and editing operations

1.1.4.1 Poligoni irregolari

Story:

Come utente,
voglio poter creare poligoni irregolari come nuove forme,
in modo da poter ideare figure personalizzate nel riquadro di disegno.

Acceptance criteria:

- Dato un poligono irregolare presente nel riquadro di disegno,
Quando lo seleziono per cliccare sul comando apposito (ad es. "rendi una forma - poligono irregolare"),
Allora questo poligono irregolare deve diventare una forma riutilizzabile.
- Data una nuova forma "poligono irregolare" inserita,
Quando la seleziono come una qualsiasi altra forma,
Allora devo poterlo manipolare come le altre forme.

Priority: Medio - Bassa

Story points: 8

1.1.4.2 Stringhe

Story:

Come utente,
voglio poter creare stringhe di testo riconosciute come forme,
in modo da manipolare le stringhe come le forme solitamente utilizzabili.

Acceptance criteria:

- Data una stringa presente nel riquadro di disegno,
Quando clicco sul comando apposito (ad es. "rendi una forma - stringa"),
Allora questa stringa deve diventare una forma riutilizzabile.
- Data una nuova forma "stringa" inserita,
Quando la seleziono,
Allora devo poterla manipolare come le altre forme.
- Data una forma di tipo "stringa" selezionata,
Quando utilizzo i controlli di modifica del testo,
Allora devo poter modificare la dimensione dei caratteri della stringa stessa.

Priority: Medio - Bassa

Story points: 5

1.1.4.3 Rotazione di una forma

Story:

Come utente,
voglio poter ruotare arbitrariamente una forma (o gruppo di forme),
in modo da poterne regolare liberamente l'angolazione, nel riquadro di disegno.

Acceptance criteria:

- Data una forma, presente nel riquadro di disegno,
Quando la seleziono,
Allora devo essere disponibile un comando, o un punto di manipolazione, apposito per ruotarla (ad es. "ruota").
- Data una forma per cui è stato selezionato il comando apposito per ruotarla, oppure per cui è sfruttato il punto di manipolazione dedito alla rotazione,
Quando muovo il mouse lungo la circonferenza (non visibile) che circonda la forma,
Allora la forma deve ruotare, seguendo in tempo reale i movimenti del cursore.

Priority: Medio - Bassa

Story points: 5

1.1.4.4 Specchiatura

Story:

Come utente,
voglio poter specchiare, orizzontalmente e verticalmente, una forma (o un gruppo di forme) selezionata,
in modo da poterne variare l'orientamento visivo.

Acceptance criteria:

- Data una forma presente nel riquadro di disegno,
Quando la seleziono,
Allora deve essere disponibile un comando apposito per specchiarla (ad es. "specchia la selezione").
- Data una forma selezionata,
Quando attivo il comando dedito alla specchiatura,
Allora la forma deve essere specchiata rispetto all'asse orizzontale, o verticale, in base all'opzione selezionata.

Priority: Medio - Bassa

Story points: 5

1.1.4.5 Stretch di una forma

Story:

Come utente,
voglio poter allungare o comprimere una forma (o un gruppo di forme) selezionata, verticalmente e orizzontalmente,
in modo da modificarne le proporzioni.

Acceptance criteria:

- Data una forma presente nel riquadro di disegno,
Quando la seleziono,
Allora deve essere possibile attivare un comando, per deformarla.

- Data una forma selezionata,
Quando utilizzo il comando apposito, modificando lunghezza e/o larghezza,
Allora la forma deve modificare la propria dimensione solo lungo l'asse desiderato.

Priority: Medio - Bassa

Story points: 5

1.1.5 Groups and shape libraries

1.1.5.1 Selezione Multipla

Story:

Come utente,
Voglio selezionare più forme contemporaneamente,
Così da poter applicare le stesse operazioni a tutte le forme selezionate.

Acceptance Criteria:

- Dato che ho selezionato due o più forme,
Quando applico una modifica ad una di esse,
Allora le modifiche vengono applicate a tutte le forme selezionate.

Priority: Bassa

Story Points: 3

1.1.5.2 Grouping

Story:

Come utente,
Voglio raggruppare le forme selezionate,
Così da poterle trattare come un'unica forma durante le interazioni future.

Acceptance Criteria:

- Dato che ho selezionato due o più forme tramite selezione multipla,
Quando clicco il pulsante "Raggruppa",
Allora posso interagire con il gruppo di oggetti come fosse un unico.

Priority: Bassa

Story Points: 5

1.1.5.3 Ungrouping

Story:

Come utente,
Voglio disaggregare le forme precedentemente raggruppate,
Così da poter trattare nuovamente ogni forma in modo indipendente.

Acceptance Criteria:

- Dato che ho selezionato una forma ottenuta raggruppando altre forme,
Quando clicco sul pulsante "Separa",
Allora ottengo di nuovo le singole forme e posso interagirvi singolarmente.

Priority: Bassa

Story Points: 3

1.1.5.4 Shape creation commands

Story:

Come utente,
Voglio definire i "Shape creation commands" salvando una forma selezionata con un nome,
Così che possa riprodurre la stessa forma in seguito, indipendentemente dalle modifiche apportate all'originale.

Acceptance Criteria:

- Dato che ho selezionato una forma, quando clicco il pulsante "Salva Shape Creation Command" e fornisco un nome, posso visualizzarlo tra le forme supportate.
- Dato che ho salvato almeno uno "Shape creation command", quando scelgo quello che mi interessa, posso inserirlo nello spazio di lavoro.

Priority: Bassa

Story Points: 13

1.1.5.5 Memorizzazione dei "Shape creation commands"

Story:

Come utente,
Voglio che i "Shape creation commands" siano salvati all'interno del drawing file,
Così che persistano quando chiudo e riapro il programma o condivido il disegno.

Acceptance Criteria:

- Dato che ho salvato almeno uno "Shape creation commands", quando clicco sul pulsante per salvare il progetto, nel drawing file vengono salvate anche i "Shape creation commands" creati.
- Dato che ho salvato un drawing file (contenente "Shape creation commands"), quando lo carico nel mio spazio di lavoro, i comandi sono disponibili per essere utilizzati.

Priority: Bassa

Story Points: 8

1.1.5.6 Esportazione di uno "Shapes library file"

Story:

Come utente,
Voglio esportare i miei "Shape creation commands" in un "Shapes library file",
Così che possa eseguirne il backup, condividerli o utilizzarli in altri disegni.

Acceptance Criteria:

- Dato che ho salvato almeno uno "Shape creation commands", quando clicco sul pulsante "Esporta Shapes Library" e scelgo un path, i comandi salvati vengono salvati in uno "Shapes library file".

Priority: Bassa

Story Points: 8

1.1.5.7 Importazione di uno "Shapes library file"

Story:

Come utente,

Voglio importare i "Shape creation commands" da un "Shapes library file" nel "drawing file" corrente,

Così che possa riutilizzare forme predefinite o librerie create da altri.

Acceptance Criteria:

- Dato che ho uno "Shape library file", quando clicco il pulsante "Import Shape Library", posso utilizzare tutte le forme presenti nel file.

Priority: Bassa

Story Points: 8

1.2 Definition of Done

1. Il codice implementa tutti i requisiti, soddisfacendo il comportamento atteso.
2. Il codice è stato testato, utilizzando casi base e avanzati, producendo output corretti.
3. Il codice è stato sottoposto a revisione incrociata da parte dei membri del team, per garantire la conformità agli standard.
4. Il modulo è stato integrato e testato insieme agli altri componenti del sistema, producendo gli output attesi senza conflitti o errori.
5. Il codice è stato integrato nel branch principale del repository, senza conflitti o errori.
6. Il codice è ben commentato, ove necessario, spiegando in modo conciso la logica e le funzionalità implementate.
7. È stata redatta una documentazione tecnica che descrive le funzionalità e include esempi utili ed espliciti sui casi d'uso delle funzionalità stesse.
8. Le funzionalità implementate rispettano i requisiti riguardanti prestazioni minime e i criteri di efficienza attesi.

1.3 Description of software architecture

In questo paragrafo è descritta l'architettura adottata per lo sviluppo del software. Questa rappresenta la struttura fondamentale, il telaio, del sistema. Inoltre, definire un'architettura permette di stabilire i componenti principali, nonché le loro responsabilità.

1.3.1 Architettura

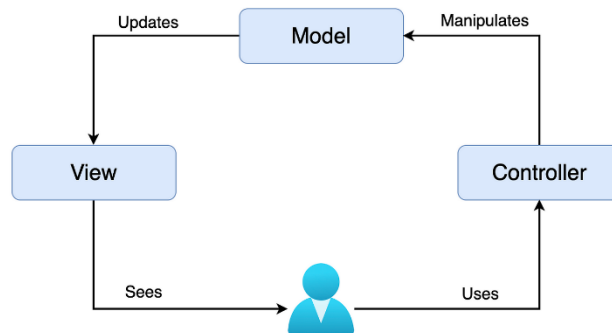


Figura 1.3.1 - Dimostrazione grafica del pattern architetturale MVC

L'applicazione verrà costruita seguendo il paradigma architetturale **Model-View-Controller** (MVC), una scelta comune per lo sviluppo di applicazioni interattive, dotate di interfaccia grafica, come nel caso del software corrente.

Questa architettura permette una netta separazione della logica di dominio, contenuta nel Model, della visualizzazione grafica, contenuta nella View, e della gestione dell'interazione utente, contenuta nel Controller.

Formalmente:

1. **Model:** è responsabile della logica applicativa, gestisce i dati e garantisce la coerenza e l'integrità dei dati (non è a conoscenza né della View né del Controller).
2. **View:** è responsabile della rappresentazione grafica delle informazioni contenute nel Model e si occupa di fornire all'utente un'interfaccia intuitiva e reattiva, che si aggiorni in risposta alle modifiche del Model.
3. **Controller:** media tra Model e View, intercettando gli input dell'utente e interpretandoli, va ad invocare le appropriate operazioni sul Model. Conseguentemente, aggiorna la View per riflettere le modifiche avvenute.

Tra i vantaggi principali di questo tipo di architettura abbiamo:

- **Manutenibilità e testabilità:** vi è una netta separazione tra logica applicativa e interfaccia, consentendo di testare e mantenere le singole componenti indipendentemente le une dalle altre.
- **Estendibilità e modularità:** l'architettura facilita l'aggiunta di nuove funzionalità, permettendo l'applicazione di modifiche localizzate alle componenti interessate senza impattare l'intero sistema, riducendo il rischio di introdurre regressioni nel sistema complessivo.
- **Collaborazione semplificata:** permette una suddivisione del lavoro più efficiente tra i membri del gruppo di lavoro, in modo chiaro. Ogni membro del team può concentrarsi su una componente specifica, in modo da evitare interferenze con il lavoro altrui.

1.4 Architecture and technologies chosen

Data la scelta di MVC come architettura abbiamo effettuato le seguenti scelte a livello di tecnologie:

- Linguaggio: Java
- GUI toolkit: JavaFX
- Testing: JUnit
- Versioning: Git + GitHub
- Gestione attività: Trello
- Development environment: NetBeans

1.5 Mock-up interface

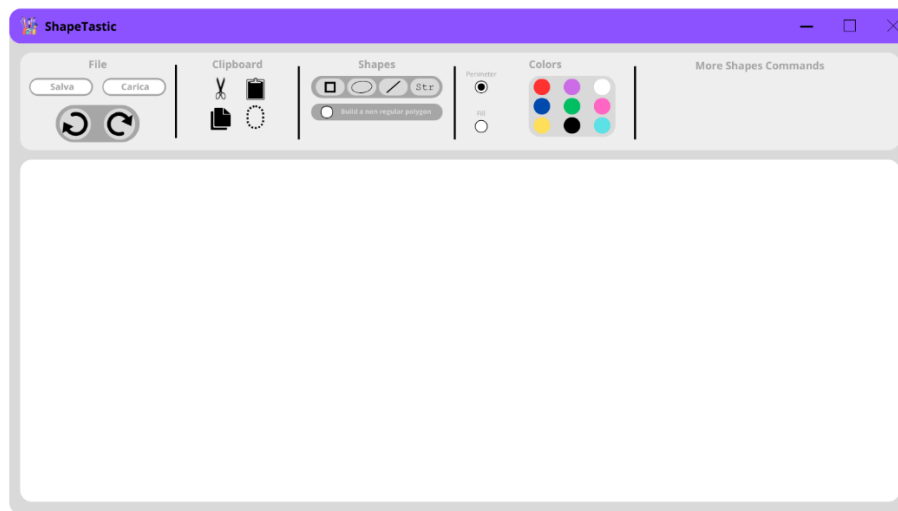


Figura 1.5.1 - Interfaccia prevista per il sistema

2 First Sprint

2.1 First sprint Planning

2.1.1 Stima iniziale della Project Velocity

In questa fase è importante definire quanti Story Points pensiamo di poter completare in una sprint. Tenendo conto che ci si aspetta che ogni membro del team lavori 8-9 ore per sprint e considerando 1 ora di lavoro per Story Point una prima stima potrebbe essere la seguente:

Velocity = 4 membri * 8-9 ore = 32-36 Story Points

Quindi la stima iniziale è di circa 32-36 Story Points per sprint.

2.1.2 First Sprint Backlog

User Story	Story Points	
Finestra iniziale	1	
Aggiunta forme geometriche	3	
Forme supportate	3	
Scelta colore del bordo delle forme	3	
Scelta colore di riempimento delle forme	3	
Salvataggio dei disegni	5	
Caricamento dei disegni	5	
Selezione forma	3	
Eliminazione di una forma	1	
Copia e incolla	3	
Taglia	2	
Totale:		32

Tabella 1: First Sprint Backlog definita nella fase Pre-Game

Abbiamo stimato che durante la prima sprint verranno implementate tutte le User Stories ad alta priorità, contenenti le basic operations, e solo una parte di quelle con priorità medio-alta, contenenti le prime operazioni di shape editing. In totale, gli story points previsti per la prima sprint sono 32.

Queste stime si tengono in linea con la stima iniziale riguardante la Project Velocity.

2.1.3 Suddivisione in task e assegnazione

2.1.3.1 Finestra iniziale (1 SP)

Obiettivo: aprire una finestra vuota all'avvio dell'app.

Tasks	Responsabile
Impostare la finestra principale	Gennaro D'Ambrosio
Inserire di un riquadro di disegno all'interno della finestra principale	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione della finestra e del riquadro di disegno	Ciro Cutolo

Tabella 2: Tasks US1

2.1.3.2 Aggiunta forme geometriche (3 SP)

Obiettivo: selezionare una forma e disegnarla con il mouse.

Tasks	Responsabile
Gestire l'occorrenza del click, sul riquadro di disegno, dopo la selezione di una forma, utile per inserire la forma selezionata	Alessio Bottiglieri
Aggiungere delle componenti logiche per creare e visualizzare una linea, una per visualizzare un rettangolo e una per visualizzare un'ellisse, nel riquadro di disegno	Ciro Cutolo
Scrivere e/o eseguire test per verificare l'effettiva presenza di una componente utile alla selezione della forma, nonché il corretto inserimento della forma nel riquadro	Gaetano Frasca

Tabella 3: Tasks US2

Nota: Questa storia dev'essere implementata dopo la 1.1.1.3 "Forme supportate".

2.1.3.3 Forme supportate (3 SP)

Obiettivo: aggiungere le forme supportate dall'applicazione.

Tasks	Responsabile
Progettare la struttura dati per rappresentare le diverse forme	Ciro Cutolo
Implementare un componente nell'interfaccia che permetta di scegliere la forma da inserire	Gaetano Frasca
Implementare la logica che colleghi il selettore di forme con la selezione effettiva della forma scelta.	Gaetano Frasca

Scrivere e/o eseguire test per verificare l'effettiva presenza di una componente utile alla visualizzazione delle tre forme supportate all'interno dell'interfaccia.	Alessio Bottiglieri
--	---------------------

Tabella 4: Tasks US3

2.1.3.4 Scelta colore bordo forma (3 SP)

Obiettivo: scegliere colore bordo prima di disegnare la forma.

Tasks	Responsabile
Aggiungere un selettore di colore per il bordo delle forme come elemento UI	Alessio Bottiglieri
Implementare un meccanismo che colleghi la scelta del colore alla forma selezionata e lo applichi al bordo della stessa al momento dell'inserimento nel riquadro di disegno	Alessio Bottiglieri
Scrivere e/o eseguire test per verificare la corretta visualizzazione di una componente UI per la selezione del colore bordo nonché il corretto funzionamento della logica implementata per modificare il colore del bordo della forma che stiamo per inserire	Gennaro D'Ambrosio

Tabella 5: Tasks US4

2.1.3.5 Scelta colore riempimento (3 SP)

Obiettivo: selezionare colore di riempimento per rettangolo o ellisse.

Tasks	Responsabile
Aggiungere un selettore di colore per la sezione interna delle forme come elemento UI	Ciro Cutolo
Implementare un meccanismo che colleghi la scelta del colore alla forma selezionata e lo applichi all'interno della stessa al momento dell'inserimento nel riquadro di disegno	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione di una componente UI per la selezione del colore di riempimento nonché il corretto funzionamento della logica implementata per modificare il colore di riempimento della forma che stiamo per inserire	Gaetano Frasca

Tabella 6: Tasks US5

2.1.3.6 Salvataggio del disegno (5 SP)

Obiettivo: Salvare il disegno in un file.

Tasks	Responsabile
Implementare la logica per convertire e salvare i disegni	Gennaro D'Ambrosio
Aggiungere un componente, dedicato al salvataggio, all'interfaccia ed implementare una finestra di dialogo utile all'utente per scegliere dove salvare il file	Gennaro D'Ambrosio
Collegare il componente dedicato al salvataggio all'effettiva funzione di salvataggio	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante di salvataggio e della finestra di dialogo per il salvataggio del file nonché il corretto funzionamento della logica di salvataggio	Ciro Cutolo

Tabella 7: Tasks US6

2.1.3.7 Caricamento del disegno (5 SP)

Obiettivo: Caricare un disegno precedentemente salvato.

Tasks	Responsabile
Scrivere la logica utile a caricare un disegno da file	Gaetano Frasca
Aggiungere un pulsante "carica" all'interfaccia ed implementare una finestra di dialogo per permettere all'utente di selezionare il file da caricare	Gaetano Frasca
Collegare il pulsante "carica" all'effettiva funzione di caricamento	Gaetano Frasca
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante di caricamento e della finestra di dialogo per il caricamento del file nonché il corretto funzionamento della logica di caricamento	Alessio Bottiglieri

Tabella 8: Tasks US7

2.1.3.8 Selezione forma (3 SP)

Obiettivo: selezionare una forma già presente nel disegno con il mouse.

Tasks	Responsabile
Implementare la logica utile a determinare quando il click del mouse ricade su di una forma	Alessio Bottiglieri
Aggiungere la logica utile a evidenziare graficamente la forma selezionata	Alessio Bottiglieri
Implementare la logica che permetta di gestire la forma selezionata, per effettuare azioni rispetto a quest'ultima (Gestione dello stato interno "forma selezionata" per abilitare operazioni successive")	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione grafica di una forma selezionata nonché della logica per gestire lo stato interno di quest'ultima	Gennaro D'ambrosio

Tabella 9: Tasks US8

2.1.3.9 Eliminazione di una forma (1 SP)

Obiettivo: eliminare una forma selezionata dal disegno

Tasks	Responsabile
Aggiungere una voce al menu, "Elimina", accanto ad una forma selezionata	Ciro Cutolo
Implementare la logica che colleghi la pressione della voce di menu apposita con l'effettiva eliminazione della forma selezionata	Ciro Cutolo
Implementare la logica che permetta di aggiornare correttamente il riquadro di disegno, dopo l'eliminazione	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione della voce "Elimina" accanto alla forma selezionata nonché della logica per aggiornare il riquadro dopo l'eliminazione	Alessio Bottiglieri

Tabella 10: Tasks US9

2.1.3.10 Copia e incolla (3 SP)

Obiettivo: copiare la forma selezionata ed incollarla in un altro punto

Tasks	Responsabile
Implementare struttura dati per rappresentare gli "appunti" (clipboard)	Alessio Bottiglieri
Implementare graficamente i pulsanti di "Copia" ed "Incolla"	Alessio Bottiglieri
Implementare la logica utile alla gestione delle funzioni "copia" e "incolla"	Alessio Bottiglieri
Creare un collegamento tra la pressione delle voci "copia" e "incolla" e la logica che permetta di azionare effettivamente i comandi	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare la corretta visualizzazione dei pulsanti di "copia" e "incolla" accanto alla forma selezionata nonché la logica per implementare questi comandi	Ciro Cutolo

Tabella 11: Tasks US10

2.1.3.11 Taglia (2 SP)

Obiettivo: Rimuovere e salvare la forma negli appunti

Tasks	Responsabile
Aggiungere una voce di menu "taglia"	Ciro Cutolo
Definire la logica utile a sfruttare contemporaneamente le funzioni "copia" ed "elimina"	Ciro Cutolo
Implementare la logica che colleghi la pressione della voce di menu con l'effettiva azione del comando "taglia"	Ciro Cutolo
Scrivere e/o eseguire test per verificare la corretta visualizzazione del pulsante "taglia" accanto alla forma selezionata nonché la logica per implementare effettivamente il comando	Gaetano Frasca

Tabella 12: Tasks US11

2.2 First Sprint – Applicazione Pattern

2.2.1 Simple Factory Pattern

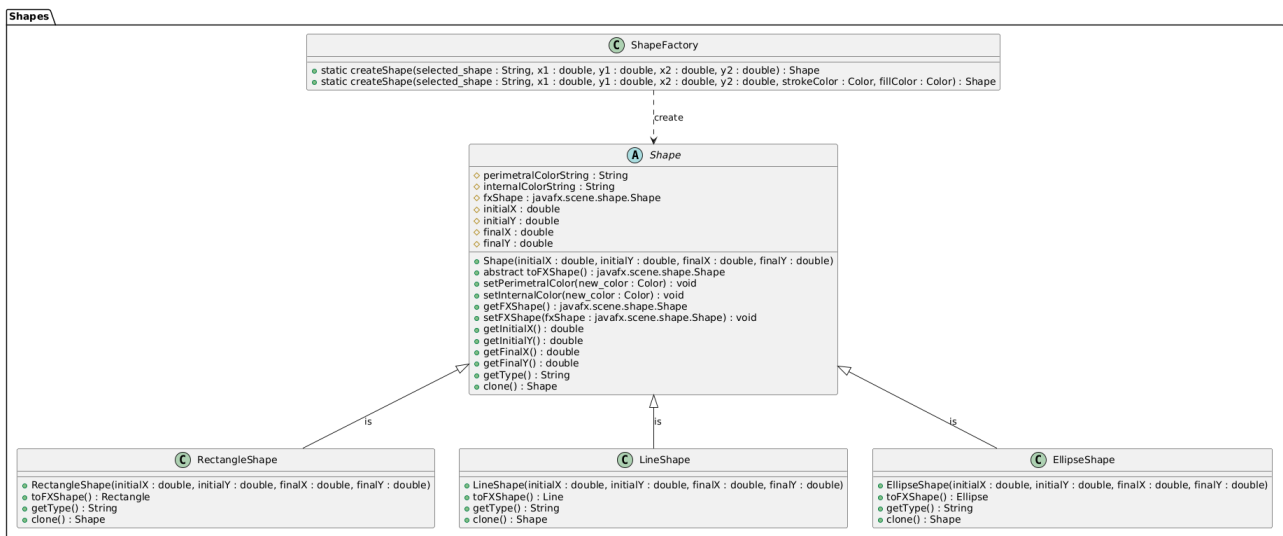


Figura 2: Simple Factory Pattern - Class Diagram

Creational Pattern utilizzato per rendere le classi indipendenti dalla logica di creazione degli oggetti che usano. Nel caso particolare viene applicato alle classi dell'architettura che andranno a rappresentare le forme supportate dal programma: linea, rettangolo ed ellisse.

- L'interfaccia *Shape* riveste il ruolo di *Product*, comune a tutti gli oggetti che possono essere creati dal *Creator*. funge da tipo comune per tutte le forme.
- Le classi concrete *LineShape*, *RectangleShape* ed *EllipseShape*, dette in generale *ConcreteProduct*, implementano l'interfaccia *Shape* per definire il comportamento specifico di ciascuna forma.
- La classe *ShapeFactory* agisce come una fabbrica, un *Creator*, incapsulando la logica di istanziazione all'interno di un unico metodo, il *factory method*, che riceve un identificatore del tipo di forma, una stringa, e restituisce il nuovo oggetto forma corrispondente.

Questo approccio semplifica la creazione degli oggetti, garantisce coerenza e permette all'applicazione di generare nuove forme senza esporre la logica di istanziazione al resto del codice.

Per questo sistema, si è scelto di applicare il Simple Factory al posto del Factory Method, perché più adatto a un'applicazione di piccole dimensioni con un numero limitato di tipi di oggetti e sufficiente per le esigenze del progetto. L'utilizzo del pattern Factory avrebbe introdotto una complessità eccessiva, richiedendo sottoclassi per ogni tipo di forma, senza apportare vantaggi significativi in questo contesto.

2.2.2 Prototype Pattern

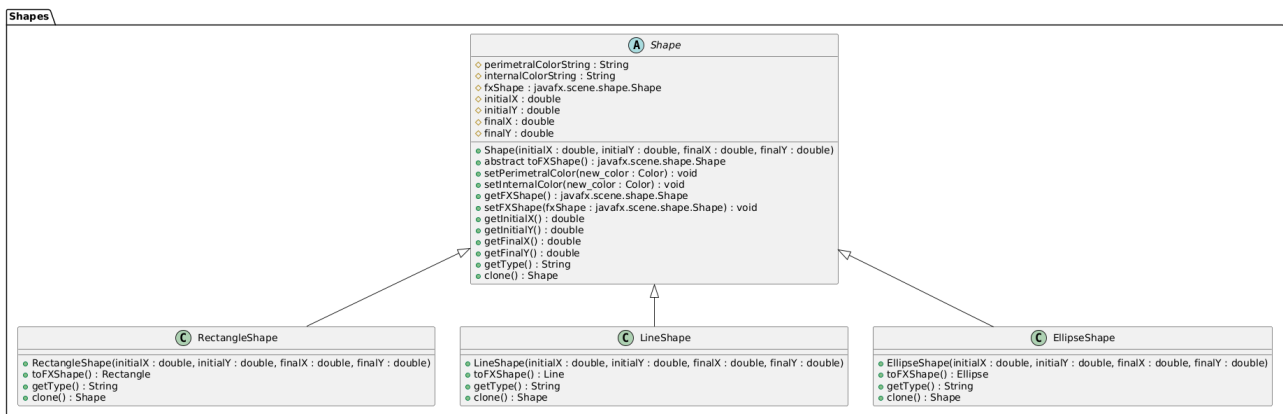


Figura 3: Prototype Pattern - Class Diagram

Il pattern Prototype è stato applicato per consentire la clonazione degli oggetti Shape all'interno dell'applicazione di disegno, in modo efficiente e flessibile. In particolare, questo pattern si rivela utile quando si ha la necessità di creare nuove istanze di oggetti complessi partendo da un oggetto esistente, anziché ricorrere ogni volta alla costruzione manuale.

Nel nostro caso, la classe astratta Shape, che riveste il ruolo di *Prototype*, implementa l'interfaccia Cloneable e fornisce un metodo clone(), rendendolo accessibile alle classi ConcretePrototype. Queste classi concrete, nel nostro caso LineShape, RectangleShape e EllipseShape, ereditano la logica di clonazione di base e di personalizzarla ove necessario. Ad esempio, LineShape ridefinisce clone() per creare una copia logica e grafica dell'oggetto originale, inclusa la duplicazione dell'oggetto JavaFX Line, che rappresenta la parte visuale della forma.

La scelta del Prototype consente di evitare la creazione ripetuta di oggetti tramite costruttori, soprattutto quando molte proprietà devono essere replicate. Inoltre, la logica di clonazione è incapsulata in ciascuna classe rappresentate una forma, rendendo il sistema facilmente estendibile: ogni nuova forma potrà semplicemente implementare la propria versione di clone().

Nel complesso, l'uso del pattern Prototype ha migliorato la modularità e la riusabilità del codice, permettendo la duplicazione rapida delle forme nel canvas senza introdurre dipendenze rigide o codice ridondante.

2.3 First Sprint – Testing

Come previsto dalla Definition of Done (DoD), tutte le funzionalità sviluppate durante la sprint sono state sottoposte a test per garantirne la correttezza e la qualità.

In particolare, sono stati effettuati:

- Unit Test sulle classi Java, al fine di verificare il comportamento corretto delle singole unità di codice in isolamento. Tali test sono presenti all'interno della repository di progetto, insieme al relativo codice sorgente.
- Functional Test sull'interfaccia utente, con l'obiettivo di validare il corretto funzionamento delle funzionalità dal punto di vista dell'utente finale. Alcuni di questi test sono documentati nelle sezioni successive.

L'esecuzione di questi test ha permesso di identificare e correggere tempestivamente eventuali anomalie, contribuendo al rispetto dei criteri di accettazione e al rilascio di incrementi di prodotto conformi agli standard qualitativi definiti.

2.4 First Sprint Backlog

Durante questa sprint, il team ha lavorato in modo efficace e collaborativo, riuscendo a completare tutti i task pianificati all'interno dello Sprint Backlog. Le user stories previste sono state tutte portate a termine e marcate come Done, in piena conformità con i criteri stabiliti nella Definition of Done. Questo risultato riflette una buona pianificazione iniziale, un'efficace suddivisione dei compiti e un costante monitoraggio dell'avanzamento durante lo sprint. Il rispetto della Definition of Done garantisce inoltre che ogni storia sia stata adeguatamente testata, documentata e integrata nel prodotto, assicurando un incremento di valore concreto e potenzialmente rilasciabile.

User Story	Story Points	Stato
Finestra iniziale	1	DONE
Aggiunta forme geometriche	3	DONE
Forme supportate	3	DONE
Scelta colore del bordo delle forme	3	DONE
Scelta colore di riempimento delle forme	3	DONE
Salvataggio dei disegni	5	DONE
Caricamento dei disegni	5	DONE
Selezione forma	3	DONE
Eliminazione di una forma	1	DONE
Copia e incolla	3	DONE

Taglia	2	DONE
--------	---	------

Tabella 13: First Sprint Backlog - Report Fine Sprint

2.5 Updated Product Backlog

Backlog ID	User Story	SP	Backlog ID	User Story	SP	Backlog ID	User Story	SP
1	Finestra iniziale	1	13	Modifica della posizione di una forma	2	25	Rotazione di una forma	5
2	Aggiunta forme geometriche	3	14	Modifica del colore di una forma	2	26	Specchiatura	5
3	Forme supportate	3	15	Modifica delle dimensioni di una forma	5	27	Stretch di una forma	5
4	Scelta colore del bordo delle forme	3	16	Modifica del livello di una forma	3	28	Selezione Multipla	3
5	Scelta colore di riempimento delle forme	3	17	Modifica del livello di zoom	5	29	Grouping	5
6	Salvataggio dei disegni	5	18	Area di disegno espandibile	3	30	Ungrouping	3
7	Caricamento dei disegni	5	19	Scroll area di disegno	2	31	Shape creation commands	13
8	Selezione forma	3	20	Visualizzazione della griglia	3	32	Memorizzazione dei "Shape creation commands"	8
9	Eliminazione di una forma	1	21	Attivazione e disattivazione della griglia	1	33	Esportazione di uno "Shapes library file"	8
10	Copia e incolla	3	22	Cambiare dimensioni griglia	3	34	Importazione di uno "Shapes library file"	8
11	Taglia	2	23	Poligoni irregolari	8			

12	Undo	5	24	Stringhe	5			
----	------	---	----	----------	---	--	--	--

Tabella 14: Product backlog aggiornato alla seconda sprint

2.6 Project Burndown chart:

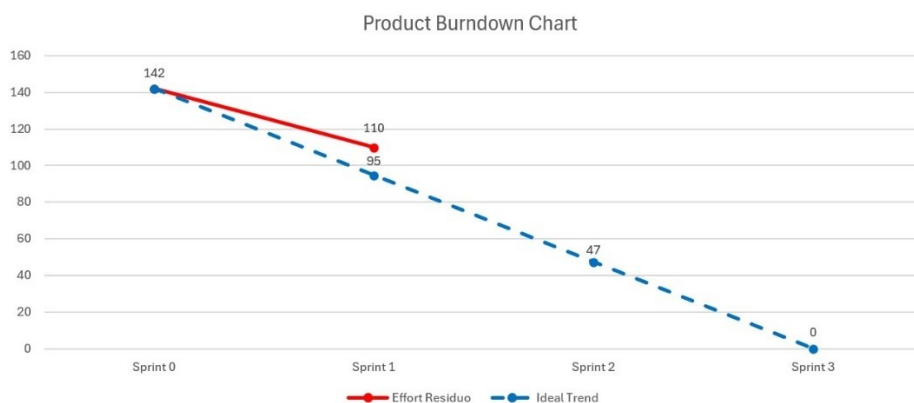


Figura 2.6.1 - Burndown Chart (First Sprint Update)

Il grafico mostra l'andamento del lavoro rimanente (espresso in story points) sprint-by-sprint. La linea tratteggiata blu rappresenta il trend ideale di completamento del progetto, mentre la linea rossa mostra l'andamento effettivo.

Al termine della prima sprint, il lavoro residuo è di 110 SP, il team ha leggermente sottoperformato rispetto alla previsione ideale (riduzione a 95 SP). Tuttavia, la velocity stimata è stata rispettata, mantenendo un ritmo di lavoro sostenibile.

Questo dato è particolarmente significativo: la velocity raggiunta può essere utilizzata come base di riferimento per la pianificazione della prossima sprint.

2.7 First Sprint - Review

2.7.1 Project velocity measured:

Durante la fase di Planning della prima Sprint era stata stimata una velocity pari a 32 SP. Alla fine della sprint tale stima risulta essere ben bilanciata, in quanto il team è riuscito a “bruciare” tutti gli SP previsti e di conseguenza a completare tutte le storie presenti nello Sprint Backlog.

La suddivisione del lavoro è stata equa, infatti, rispetto ai task previsti, i componenti della squadra hanno adempiuto ad una media di 8 story points, cadauno.

Nome Membro	Story points
Alessio Bottiglieri	8.15
Ciro Cutolo	8.45
Gaetano Frasca	7.70
Gennaro D'Ambrosio	7.70

Tabella 15: Suddivisione SP tra i membri del team - 1st Sprint

Nota: la suddivisione degli story points è stata effettuata rispetto ai task e alla loro importanza nel contesto della storia da cui sono stati determinati.

2.7.2 Problemi riscontrati durante l'implementazione

Nel corso della sprint non sono emersi technical debt da portare al prossimo sprint, in quanto tutti gli errori e i problemi sono stati individuati e risolti grazie alla cross review applicata all'intero codice. Questo processo di revisione incrociata ha permesso di mantenere alta la qualità del software e di intervenire tempestivamente su eventuali criticità. Se dovessimo individuare una funzionalità particolarmente impegnativa, sarebbe l'implementazione dell'inserimento delle forme sul canvas, che ha richiesto un notevole investimento di tempo soprattutto nella fase iniziale dello sprint

2.7.3 Modifiche da apportare al Product Backlog

Durante questo sprint non è stato necessario correggere o modificare in modo significativo le user story, il che dimostra che la fase di planning iniziale e la fase di backlog grooming sono state eseguite correttamente. Le user story erano chiare, ben definite e in linea con la comprensione del team fin dall'inizio. Tuttavia, il lavoro svolto durante questo sprint sarà prezioso per quello successivo, in particolare per aiutarci a definire e raffinare meglio le user story a grana grossa previste per i prossimi sviluppi.

2.8 First Sprint - Retrospective

Durante la Sprint Retrospective, il team ha utilizzato lo Starfish Diagram come strumento visivo e collaborativo per riflettere sull'andamento della sprint conclusa. Questo approccio ha permesso di raccogliere in modo strutturato opinioni, esperienze e suggerimenti da parte di tutti i membri del team, facilitando una discussione costruttiva.

2.8.1 Starfish Diagram

2.8.1.1 Stop (things to stop doing):

- Perdere tempo su dettagli secondari prima di completare le funzionalità principali.
- Procrastinare la scrittura dei test automatici fino a fine implementazione.
- Modificare l'architettura del sistema, senza comunicarlo
- Modificare il codice altrui, senza preavviso

2.8.1.2 Less of:

- Comunicazione frammentata sui canali informali
- Lavoro in parallelo su task fortemente dipendenti

2.8.1.3 Keep doing:

- Suddivisione equa dei task tra i membri del team.
- Rispetto della velocity prevista e delle stime iniziali.
- Uso efficace della GitHub repository e Trello board.

2.8.1.4 More of (more things to do):

- Scrivere commenti e documentazione progressiva, durante lo sviluppo.
- Comunicare in maniera esplicita i cambiamenti che si reputano necessari, anche se minimi.

2.8.1.5 Start (things to start doing):

- Tenere traccia degli imprevisti per migliorare la stima nei prossimi sprint.

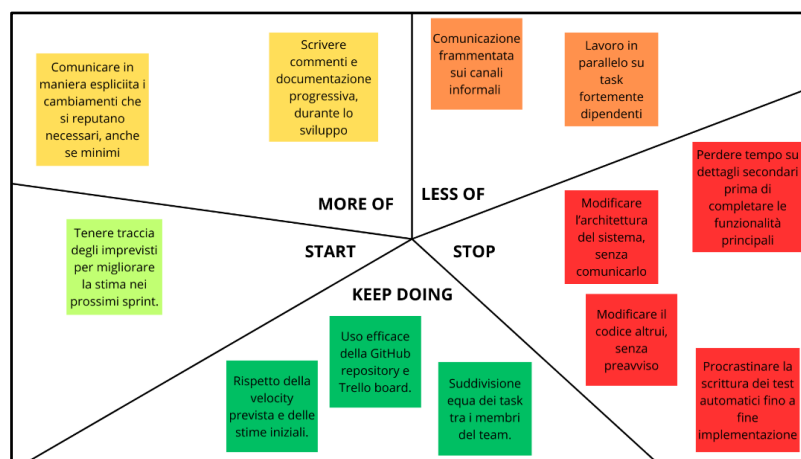


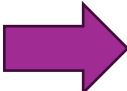
Figura 2: Starfish Diagram - 1st Sprint

3 Second sprint

3.1 Second Sprint Planning

3.1.1 Modifiche Product Backlog

Durante questo sprint, si è reso necessario rivedere il Product Backlog sulla base delle conoscenze acquisite dal lavoro completato nella prima sprint. In particolare, la User Story #16 ("Modifica del livello di una forma") è stata rivalutata e i suoi SP sono stati aumentati da 3 a 5, poiché l'implementazione ha rivelato un livello di complessità superiore a quello stimato inizialmente.

Backlog ID	User Story	SP		Backlog ID	User Story	SP
16	Modifica del livello di una forma	3		16	Modifica del livello di una forma	5

Inoltre, il team ha individuato l'opportunità di migliorare la chiarezza e la coerenza delle storie utente correlate. In particolare, la User Story #21, che riguardava l'attivazione e la disattivazione della griglia, si sovrapponeva in modo significativo alla #22, incentrata sulla funzionalità di visualizzazione della griglia. Data la loro stretta relazione e per ridurre la ridondanza, abbiamo deciso di consolidare queste due storie. La User Story #21 è stata rimossa e la descrizione e i criteri di accettazione della User Story #22 ("Visualizzazione della griglia") sono stati aggiornati per incorporare le funzionalità di attivazione e disattivazione. Questa modifica garantisce un backlog più snello e riflette una rappresentazione più accurata dell'interazione dell'utente con il componente griglia.

Di seguito è riportata la nuova User Story #21 ("Visualizzazione della griglia"):

Story:

Come utente,
voglio poter visualizzare una griglia sull'area di disegno,
così da posizionare più facilmente le forme.

Acceptance criteria:

- Data una sezione dell'interfaccia per il controllo della griglia,
Quando l'utente visualizza la sezione interessata,
Allora è presente un pulsante che permette l'attivazione e disattivazione della griglia
- Data l'area di disegno,
Quando l'utente attiva la visualizzazione della griglia,
Allora vengono mostrate linee orizzontali e verticali equidistanti, posizionate sopra lo sfondo ma sotto le forme.

- Data una griglia visibile,
Quando l'utente interagisce con le forme (es. selezione o trascinamento),
Allora le linee della griglia non interferiscono con tali operazioni.
- Data la visualizzazione della griglia attivata,
Quando l'area di disegno viene aggiornata (es. tramite scroll o zoom),
Allora la griglia si muove coerentemente con la canvas e adatta la distanza tra le linee proporzionalmente al livello di zoom.
- Data l'area di disegno,
Quando l'utente disattiva la visualizzazione della griglia,
Allora le linee della griglia scompaiono senza modificare le forme presenti.

Priority: Media

Story points: 3

3.1.2 Stima aggiornata della Project Velocity

Dati i risultati ottenuti nella prima sprint, è stato ritenuto opportuno incrementare la velocity prevista per la sprint corrente, fissandola a 35 story points.

3.1.3 Selezione delle User Stories per la seconda sprint

User Story	Story Points
Undo	5
Modifica della posizione di una forma	2
Modifica del colore di una forma	2
Modifica delle dimensioni di una forma	5
Modifica del livello di una forma	5
Modifica del livello di zoom	5
Area di disegno espandibile	3
Scroll area di disegno	2
Visualizzazione della griglia	3
Cambiare dimensioni griglia	3
Totale:	
35	

Tabella 16 - Second sprint backlog

Durante la seconda sprint verranno implementate tutte le User Stories di priorità medio-alta/media, contenenti le rimanenti *shape editing operations*, e tutte le storie legate alla sezione *display of the drawing*. In totale, gli story points previsti per la seconda sprint sono '35'. Queste stime eccedono la velocity stimata per la prima sprint, poiché è stato ritenuto possibile incrementare il lavoro, rispetto al tempo a disposizione.

3.1.4 Suddivisione in task e assegnazione

3.1.4.1 Undo (5 SP)

Obiettivo: permettere l'annullamento delle azioni effettuate sull'area di disegno

Tasks	Responsabile
Inserire un componente di interfaccia, utile ad azionare il comando undo	Ciro Cutolo
Implementare la logica dedita all'inversione delle modifiche apportate al riquadro di disegno	Ciro Cutolo
Implementare la logica che colleghi la pressione del pulsante utile per effettuare l'undo, all'effettiva azione del comando	Ciro Cutolo
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente undo	Gaetano Frasca

Tabella 17: US12

3.1.4.2 Modifica della posizione di una forma (2 SP)

Obiettivo: Consentire all'utente di trascinare una forma selezionata per modificarne la posizione all'interno del canvas

Tasks	Responsabile
Implementare la logica che permette di muovere una forma, dopo averla selezionata	Gennaro D'Ambrosio
Scrivere e/o eseguire i test per accertare la possibilità di muovere le forme	Alessio Bottiglieri

Tabella 18: Tasks US13

3.1.4.3 Modifica del colore di una forma (2 SP)

Obiettivo: Modifica del colore di contorno e riempimento di una forma già presente sulla canvas.

Tasks	Responsabile
Implementare la logica utile a cambiare il colore di una forma selezionata	Alessio Bottiglieri
Scrivere e/o eseguire i test atti a verificare l'effettiva possibilità di cambiare il colore di forme già presenti nel riquadro di disegno	Gennaro D'Ambrosio

Tabella 19: Tasks US14

3.1.4.4 Modifica delle dimensioni di una forma (5 SP)

Obiettivo: Ridimensionamento libero delle forme nel canvas.

Tasks	Responsabile
Implementare la logica utile a modificare le dimensioni di una forma, una volta selezionata	Gaetano Frasca
Scrivere e/o eseguire i test per verificare la possibilità di modificare le dimensioni di una forma già presente nel riquadro di disegno	Ciro Cutolo

Tabella 20: Tasks US15

3.1.4.5 Modifica del livello di una forma (5 SP)

Obiettivo: Gestione del livello di sovrapposizione delle forme.

Tasks	Responsabile
Implementare una struttura di livelli di disegno	Alessio Bottiglieri
Implementare la logica utile a inserire un componente del disegno in uno dei livelli liberi	Alessio Bottiglieri
Implementare la logica utile a modificare il livello di una forma già presente nel riquadro di disegno	Alessio Bottiglieri
Scrivere e/o eseguire il test per verificare che sia possibile spostare un elemento, presente nel riquadro di disegno, in un diverso livello di visualizzazione	Ciro Cutolo

Tabella 21: Tasks US16

3.1.4.6 Modifica del livello di zoom (5 SP)

Obiettivo: Controllo dello zoom sull'area di disegno.

Tasks	Responsabile
Inserire un componente di interfaccia, utile a modificare il livello di zoom	Gennaro D'Ambrosio
Implementare la logica dedicata alla modifica del livello di zoom	Gennaro D'Ambrosio
Implementare la logica che colleghi la pressione del pulsante utile per modificare il livello di zoom all'effettiva azione del comando	Gennaro D'Ambrosio

Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente di zoom	Gaetano Frasca
---	----------------

Tabella 22: Tasks US17

3.1.4.7 Area di disegno espandibile (3 SP)

Obiettivo: Canvas espandibile oltre i limiti della finestra visibile.

Tasks	Responsabile
Implementare un riquadro di disegno espandibile e scrollabile	Ciro Cutolo
Implementare un meccanismo che permetta di adattare dinamicamente le dimensioni del riquadro di disegno, considerando anche forme al di fuori della zona visibile	Ciro Cutolo
Scrivere e/o eseguire test per accertare il corretto funzionamento del riquadro espandibile	Gennaro D'Ambrosio

Tabella 23: Tasks US18

3.1.4.8 Scroll area di disegno (2 SP)

Obiettivo: Scorrimento dell'area di disegno per navigare nei contenuti.

Tasks	Responsabile
Inserire un componente di interfaccia utile a muoverci all'interno dell'area di disegno	Ciro Cutolo
Implementare la logica dedicata al movimento all'interno dell'area di disegno	Ciro Cutolo
Implementare la logica che colleghi l'interazione con il componente dedicato a muoverci all'interno dell'area di disegno con l'effettiva azione del comando	Ciro Cutolo
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente per muoverci all'interno dell'area di disegno	Alessio Bottiglieri

Tabella 24: Tasks US19

3.1.4.9 Visualizzazione della griglia (3 SP)

Obiettivo: Visualizzazione opzionale della griglia sul canvas.

Tasks	Responsabile
Inserire un componente di interfaccia utile ad attivare e disattivare la griglia nell'area di disegno	Gaetano Frasca

Implementare la logica dedicata all'inserimento e alla rimozione della griglia nell'area di disegno	Gaetano Frasca
Implementare la logica che colleghi l'interazione con il componente grafico utile all'attivazione e alla disattivazione della griglia con l'effettiva azione dei comandi	Gaetano Frasca
Scrivere e/o eseguire test che verifichino il corretto funzionamento dei componenti per l'inserimento e l'eliminazione della griglia dall'area di disegno	Gennaro D'Ambrosio

Tabella 25: Tasks US20

3.1.4.10 Cambiare dimensioni griglia (3 SP)

Obiettivo: Personalizzazione della dimensione della griglia

Tasks	Responsabile
Inserire un componente di interfaccia, utile a modificare la dimensione della griglia	Alessio Bottiglieri
Implementare la logica dedicata alla modifica della dimensione della griglia	Alessio Bottiglieri
Implementare la logica che colleghi la pressione del pulsante utile per modificare la dimensione della griglia all'effettiva azione del comando	Alessio Bottiglieri
Scrivere e/o eseguire test che verifichino il corretto funzionamento del componente per la modifica della dimensione della griglia	Gaetano Frasca

Tabella 26: Tasks US21

3.2 Second Sprint – Applicazione Pattern

3.2.1 Command Pattern

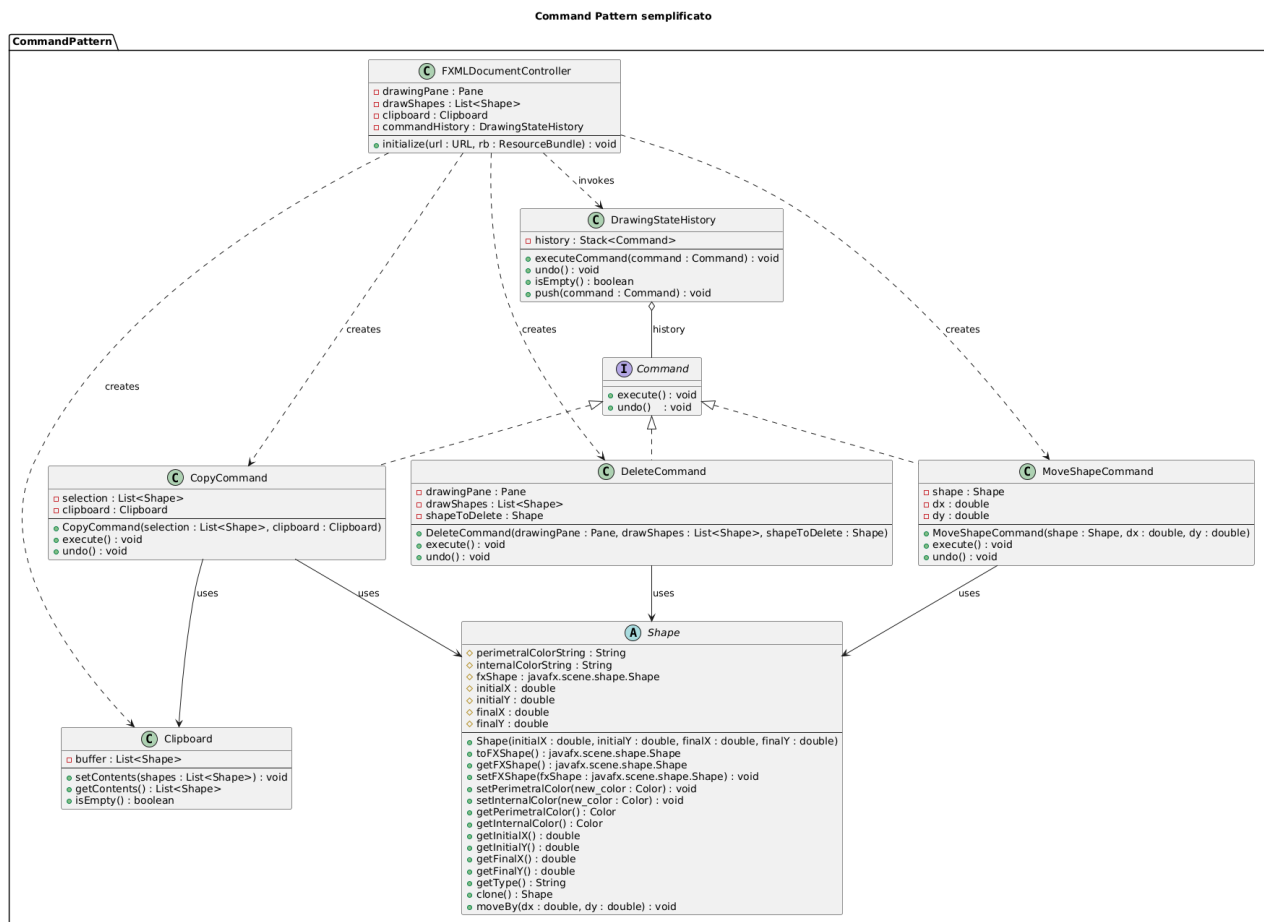


Figura 3: Command Pattern – Class Diagram

Nota: per semplicità sono state incluse solo tre classi concrete. Inoltre sono stati mostrati solo gli attributi più importanti del Controller

Il Command è un pattern comportamentale che permette di incapsulare un'azione sotto forma di oggetto, rendendo l'operazione (e il suo annullamento) indipendente dall'interfaccia utente che la invoca. Nella nostra applicazione di disegno è stato adottato questo pattern per realizzare la funzionalità di Undo e, più in generale, per uniformare la gestione di tutte le operazioni che l'utente può compiere sul canvas.

- L'interfaccia Command rappresenta il ruolo di *Command* nel pattern e definisce i metodi *execute()* e *undo()* che tutte le operazioni concrete devono implementare.
- Le classi concrete *InsertShapeCommand*, *DeleteCommand*, *MoveShapeCommand*, *ChangeColorCommand*, *CopyCommand*, *PasteCommand*, *BringForwardCommand*, *BringToFrontCommand*, *SendBackwardCommand*, *SendToBackCommand* (dette *ConcreteCommand*) implementano l'interfaccia *Command* e incapsulano singole operazioni sul canvas, mantenendo al loro interno lo stato necessario per permettere l'operazione di annullamento (*undo()*).

- La classe *DrawingStateHistory* riveste il ruolo di *Invoker*, gestendo l'esecuzione dei comandi tramite il metodo *executeCommand(command)* e mantenendo lo storico delle operazioni per abilitare l'Undo.
- Le classi *Shape* e *Clipboard*, insieme agli oggetti *Pane drawingPane* e *List<Shape> drawShapes* gestiti direttamente dal controller, svolgono il ruolo di *Receiver*. Questi componenti eseguono concretamente le operazioni richieste dai *ConcreteCommand*, come lo spostamento delle forme, la modifica del colore, l'inserimento o la cancellazione nel canvas, le operazioni di copia/incolla ecc.
- Il *FXMLDocumentController* agisce da Client: interpreta le azioni dell'utente, istanzia e configura i comandi concreti e li passa all'Invoker per l'esecuzione.

L'alternativa, cioè gestire ogni azione direttamente nel controller, avrebbe comportato

- Logica di undo replicata per ogni caso d'uso
- Codice del controller monolitico, poco manutenibile e difficilmente testabile

Il *Command* pattern permette invece una struttura modulare, in linea con il *Single Responsibility Principle*: ogni comando si occupa di una sola cosa, e la gestione della memoria della cronologia dei comandi è centralizzata. Questo ha migliorato la coesione del progetto e, insieme ai pattern *Prototype* e *Simple Factory*, ha reso il codice facilmente estendibile e privo di dipendenze rigide.

3.3 Second Sprint Backlog

Durante questa sprint, il team ha lavorato in modo efficace e collaborativo, riuscendo a completare tutti i task pianificati all'interno dello Sprint Backlog. Le user stories previste sono state tutte portate a termine e marcate come Done, in piena conformità con i criteri stabiliti nella Definition of Done. Questo risultato riflette una buona pianificazione iniziale, un'efficace suddivisione dei compiti e un costante monitoraggio dell'avanzamento durante lo sprint. Il rispetto della Definition of Done garantisce inoltre che ogni storia sia stata adeguatamente testata, documentata e integrata nel prodotto, assicurando un incremento di valore concreto e potenzialmente rilasciabile.

ID	User Story	Story Points	Stato
12	Undo	5	DONE
13	Modifica della posizione di una forma	2	DONE
14	Modifica del colore di una forma	2	DONE
15	Modifica delle dimensioni di una forma	5	DONE
16	Modifica del livello di una forma	5	DONE
17	Modifica del livello di zoom	5	DONE
18	Area di disegno espandibile	3	DONE

19	Scroll area di disegno	2	DONE
20	Visualizzazione della griglia	3	DONE
21	Cambiare dimensioni griglia	3	DONE

Tabella 27: Second Sprint Backlog - Report Fine Sprint

3.4 Updated Product Backlog

Backlog ID	User Stories	SP	Backlog ID	User Stories	SP	Backlog ID	User Stories	SP
1	Finestra iniziale	1	12	Undo	5	23	Stringhe	5
2	Aggiunta forme geometriche	3	13	Modifica della posizione di una forma	2	24	Rotazione di una forma	5
3	Forme supportate	3	14	Modifica del colore di una forma	2	25	Specchiatura	5
4	Scelta colore del bordo delle forme	3	15	Modifica delle dimensioni di una forma	5	26	Stretch di una forma	5
5	Scelta colore di riempimento delle forme	3	16	Modifica del livello di una forma	5	27	Selezione Multipla	3
6	Salvataggio dei disegni	5	17	Modifica del livello di zoom	5	28	Grouping	5
7	Caricamento dei disegni	5	18	Area di disegno espandibile	3	29	Ungrouping	3
8	Selezione forma	3	19	Scroll area di disegno	2	30	Shape creation commands	13

9	Eliminazione di una forma	1	20	Visualizzazione della griglia	3	31	Memorizzazione dei "Shape creation commands"	8
10	Copia e incolla	3	21	Cambiare dimensioni griglia	3	33	Esportazione di uno "Shapes library file"	8
11	Taglia	2	22	Poligoni irregolari	8	34	Importazione di uno "Shapes library file"	8

Tabella 28: Product backlog aggiornato al secondo sprint

3.5 Project Burndown chart:

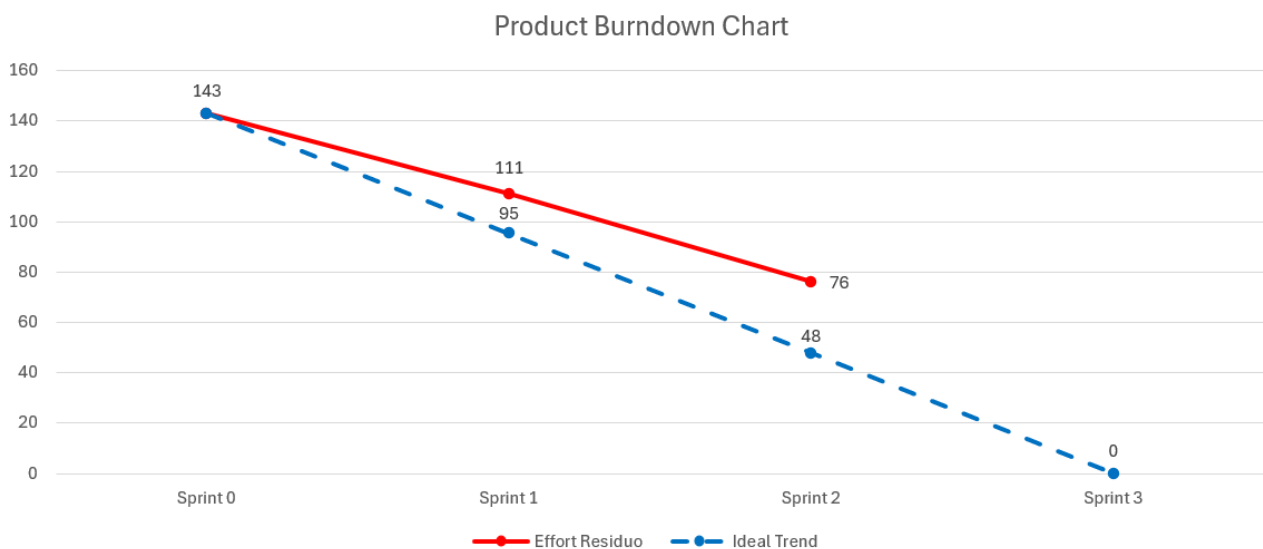


Figura 3.5.1 - Burndown chart (Second Sprint Update)

Il grafico mostra l'andamento del lavoro rimanente (espresso in story points) sprint-by-sprint. La linea tratteggiata blu rappresenta la tendenza ideale di completamento del progetto, mentre la linea rossa mostra l'andamento effettivo.

Al termine della seconda sprint, il lavoro residuo è di 76 SP. Da questo dato, si può evincere che il team ha leggermente sottoperformato rispetto alla previsione ideale (fissa a 48 SP). Tuttavia, la velocity stimata (35 SP) è stata rispettata, mantenendo un ritmo di lavoro sostenibile.

Questo dato è particolarmente significativo: la velocity raggiunta può essere utilizzata come base di riferimento per la pianificazione della prossima sprint.

3.6 Second Sprint - Review

3.6.1 Project velocity measured

Durante la fase di Planning della prima Sprint era stata stimata una velocity pari a 35 SP. Alla fine della sprint tale stima risulta essere ben bilanciata, in quanto il team è riuscito a “bruciare” tutti gli SP previsti e di conseguenza a completare tutte le storie presenti nello Sprint Backlog.

La suddivisione del lavoro è stata equa, infatti, rispetto ai task previsti, i componenti della squadra hanno adempiuto ad una media di 9 story points, cadauno.

Nome Membro	Story points
Alessio Bottiglieri	9
Ciro Cutolo	9
Gaetano Frasca	8
Gennaro D'Ambrosio	9

Tabella 29: Suddivisione SP tra i membri del team - 2nd Sprint

Nota: la suddivisione degli story points è stata effettuata rispetto ai task e alla loro importanza nel contesto della storia da cui sono stati determinati.

3.6.2 Problemi riscontrati durante l'implementazione

Come avvenuto per la Prima Sprint, anche la seconda è stata completata senza accumulare *Technical Debt* o lasciare bug irrisolti per l'iterazione successiva. Questo risultato è stato ottenuto grazie a stime equilibrate, a test approfonditi e a un alto livello di controllo sul codice attraverso una continua revisione incrociata. Tuttavia, una funzionalità particolarmente impegnativa da realizzare durante questa sprint, è stato il ridimensionamento dello spazio di lavoro. Questo compito ha richiesto lo sforzo collettivo di tutto il team per risolvere i vari problemi e bug che si sono presentati, soprattutto a causa dell'ulteriore complessità di garantire che la griglia si adattasse dinamicamente alle nuove dimensioni del canvas.

3.6.3 Modifiche da apportare al Product Backlog

Durante questo sprint, sono state apportate alcune modifiche al Product Backlog sulla base delle conoscenze acquisite dallo sviluppo in corso. In particolare, è stata rivalutata la complessità della User Story #3, che ha portato a un aumento dei suoi punti storia da 3 a 5. Inoltre, le User Story #21 (attivazione/disattivazione della griglia) e #22 (visualizzazione della griglia) sono risultate sovrapposte, quindi sono state consolidate eliminando la prima e aggiornando la descrizione e i criteri di accettazione della seconda per riflettere entrambe le funzionalità. Queste modifiche riflettono una comprensione più accurata della portata e delle interdipendenze dei compiti e dimostrano la reattività del team al feedback iterativo e al continuo perfezionamento del backlog.

3.7 Second Sprint - Retrospective

Durante la Sprint Retrospective, il gruppo ha utilizzato lo "Starfish Diagram", come struttura collaborativa per valutare le pratiche del team e identificare le aree di miglioramento continuo.

Durante la Seconda Sprint, il team ha fatto consapevolmente riferimento al diagramma realizzato durante l'interazione precedente, che è stato utilizzato come promemoria delle migliori pratiche concordate e delle insidie comuni da evitare. Questo riferimento visivo si è rivelato particolarmente efficace nel promuovere un comportamento coerente ed efficace del team, e nel mantenere un elevato standard di collaborazione, soprattutto nell'area della comunicazione, che era stata precedentemente identificata come un'area chiave di crescita.

Il diagramma seguente riflette i progressi del team in diverse aree e conferma che molte delle lezioni dello sprint precedente sono state interiorizzate con successo.

3.7.1 Starfish Diagram

3.7.1.1 Stop (things to stop doing):

- Applicare pattern, senza seguire fedelmente la teoria associata.

3.7.1.2 Less of:

- Redigere troppa documentazione.

3.7.1.3 Keep doing:

- Suddivisione equa dei task tra i membri del team.
- Rispetto della velocity prevista e delle stime iniziali.
- Testing incrociato del sistema e delle sue funzionalità.
- Scrivere commenti e documentazione progressiva, durante lo sviluppo.

3.7.1.4 More of (more things to do):

- Rispettare gli orari e l'organizzazione riguardo i daily meetings.
- Comunicazione efficace e periodica.

3.7.1.5 Start (things to start doing):

- Segnalare bug e difficoltà riscontrate, anche se risolte.

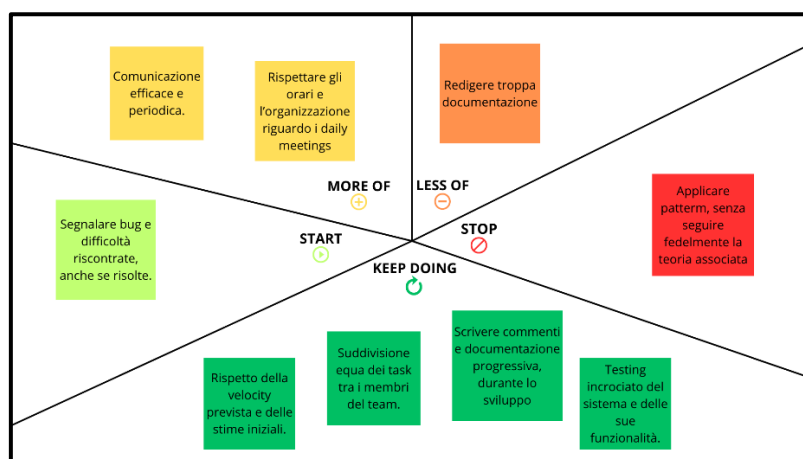


Figura 2: Starfish Diagram - Second Sprint

4 Third sprint

4.1 Third Sprint Planning

4.1.1 Modifiche Product Backlog

Durante la fase di pianificazione della Terza Sprint, il team ha provveduto a rivedere e aggiornare il Product Backlog in seguito a una segnalazione pervenuta riguardo a un malfunzionamento riscontrato nella seconda release dell'applicazione, bug non rilevato durante le attività di testing condotte nella seconda iterazione. Di conseguenza, è stata inserita una nuova user story, a massima priorità, specificamente dedicata alla risoluzione del problema e alla successiva esecuzione di test mirati, con l'obiettivo di garantire la stabilità e l'affidabilità del rilascio successivo.

Viene di seguito riportata la User Story #34 aggiunta:

Come utente,
Voglio che se applico una qualunque operazione ad una forma,
Allora deve essere possibile selezionare ed applicare altre operazioni

Acceptance Criteria

- Data una qualsiasi operazione effettuata rispetto ad una forma,
Quando seleziono la suddetta forma e faccio clic destro sulla stessa,
Allora devo poter correttamente visualizzare il menu contestuale.

4.1.2 Stima aggiornata della Project Velocity

In fase di pianificazione della terza sprint, il team ha effettuato una valutazione attenta della velocity sulla base dei risultati ottenuti nelle due iterazioni precedenti. Considerando l'incremento di efficienza riscontrato nella seconda sprint rispetto alla prima e tenendo conto della necessità di gestire una nuova user story relativa alla risoluzione di un bug emerso nella seconda release, si è deciso di aumentare ulteriormente la velocity stimata a 42 SP. Tale scelta è risultata coerente con l'obiettivo strategico di completare tutte le user story caratterizzate da priorità medio-bassa, includendo alcune a priorità bassa, al fine di consolidare le funzionalità principali dell'applicazione.

4.1.3 Selezione delle User Stories per la seconda sprint

Di seguito è riportata la Sprint Backlog relativa alla terza iterazione, composta dalle user story selezionate durante la fase di planning:

User Story	Story Points	
Bug fix	3	
Poligoni irregolari	8	
Stringhe	5	
Rotazione di una forma	5	
Specchiatura	5	
Stretch di una forma	5	
Selezione multipla	3	
Grouping	5	
Ungrouping	3	
Totale:		42

Tabella 30 - Third sprint backlog

Durante la terza sprint verranno implementate tutte le User Stories di priorità medio-bassa e alcune di priorità bassa, inerenti alle "More shapes and editing operations", nonché alle "Groups and shapes libraries".

4.1.4 Suddivisione in task e assegnazione

4.1.4.1 Bugfix (3 SP)

Obiettivo: Risolvere il bug segnalato

Tasks	Responsabile
Risoluzione Bug	Gaetano Frasca
Verifica corretto funzionamento dell'applicazione dopo il bug fixing	Alessio Bottiglieri

Tabella 31: US34

4.1.4.2 Poligoni irregolari (8 SP)

Obiettivo: Permettere la creazione e l'utilizzo di poligoni irregolari come forme personalizzate manipolabili all'interno del riquadro di disegno.

Tasks	Responsabile
Inserire un componente di interfaccia, utile a selezionare poligoni irregolari come forma da inserire nel riquadro di disegno	Ciro Cutolo

Implementare la logica che consenta, una volta selezionato il componente dell'interfaccia, di inserire la forma nell'area di disegno .	Ciro Cutolo
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato all'utilizzo dei poligoni irregolari come forma supportata.	Gaetano Frasca

Tabella 32: Tasks US22

4.1.4.3 Stringhe (5 SP)

Obiettivo: Permettere la creazione di stringhe di testo come forme manipolabili, con possibilità di modificarne dimensione e aspetto all'interno del riquadro di disegno.

Tasks	Responsabile
Inserire un componente di interfaccia, utile a selezionare una stringa come elemento da inserire nel riquadro di disegno	Alessio Bottiglieri
Implementare la logica che consenta, una volta selezionato il componente dell'interfaccia, di inserire una stringa nell'area di disegno .	Alessio Bottiglieri
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato all'utilizzo delle stringhe come forma supportata.	Ciro Cutolo

Tabella 33: Tasks US23

4.1.4.4 Rotazione di una forma (5 SP)

Obiettivo: Consentire la rotazione arbitraria delle forme (o gruppi di forme) all'interno del riquadro di disegno tramite comandi o punti di manipolazione dedicati.

Tasks	Responsabile
Aggiunta di un componente grafico per effettuare la rotazione di una forma (o gruppo di forme).	Gennaro D'Ambrosio
Aggiunta della logica per il funzionamento della rotazione di una forma (o gruppo di forme).	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato alla rotazione di una forma (o di un gruppo di forme).	Gaetano Frasca

Tabella 34: Tasks US24

4.1.4.5 Specchiatura (5 SP)

Obiettivo: Consentire la specchiatura orizzontale e verticale delle forme (o gruppi di forme) selezionate per variarne l'orientamento visivo all'interno del riquadro di disegno.

Tasks	Responsabile
Inserire un componente dell'interfaccia, utile ad applicare l'effetto di "specchiatura" ad una stringa	Gennaro D'Ambrosio
Implementare la logica che consenta, tramite apposito componente dell'interfaccia, di applicare l'effetto di "specchiatura" ad una stringa.	Gennaro D'Ambrosio
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato all'applicazione dell'effetto di "specchiatura" ad una stringa inserita nel riquadro di disegno	Alessio Bottiglieri

Tabella 35: Tasks US25

4.1.4.6 Stretch di una forma (5 SP)

Obiettivo: Consentire l'allungamento o la compressione di una forma (o gruppo di forme) lungo gli assi orizzontale e verticale per modificarne proporzioni e dimensioni.

Tasks	Responsabile
Inserire un componente di interfaccia, utile a "deformare" una forma.	Alessio Bottiglieri
Implementare la logica che consenta, tramite il componente dedicato a tale compito, di deformare una forma già presente nel riquadro di disegno	Alessio Bottiglieri
Scrivere e/o eseguire test per verificare l'effettivo funzionamento del componente UI e della logica dedicata allo stretching delle forme	Gennaro D'Ambrosio

Tabella 36: Tasks US26

4.1.4.7 Selezione multipla (3 SP)

Obiettivo: Consentire la selezione simultanea di più forme per applicare modifiche collettive nel riquadro di disegno.

Tasks	Responsabile
Inserire la logica utile per implementare la selezione di molteplici forme.	Ciro Cutolo

Aggiungere un elemento per la visualizzazione grafica delle forme selezionate.	Ciro Cutolo
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato alla soluzione multipla delle forme.	Alessio Bottiglieri

Tabella 37: Tasks US27

4.1.4.8 Grouping (5 SP)

Obiettivo: Permettere il raggruppamento di più forme selezionate in un'unica entità manipolabile come una singola forma.

Tasks	Responsabile
Aggiungere un elemento grafico per effettuare il grouping delle varie forme selezionate.	Gaetano Frasca
Aggiungere la logica per il corretto funzionamento del grouping.	Gaetano Frasca
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato al grouping delle forme selezionate.	Ciro Cutolo

Tabella 38: Tasks US28

4.1.4.9 Ungrouping (3 SP)

Obiettivo: Consentire la separazione di un gruppo di forme precedentemente raggruppate per ripristinarne l'indipendenza.

Tasks	Responsabile
Aggiungere un elemento grafico per effettuare l' ungrouping della forma selezionata.	Gaetano Frasca
Aggiungere la logica per effettuare l' ungrouping della forma selezionata.	Gaetano Frasca
Scrivere e/o eseguire test per verificare l'effettiva presenza di un componente dedicato all' ungrouping della forma selezionata.	Gennaro D'Ambrosio

Tabella 39: Tasks US29

4.2 Third Sprint - Applicazione Pattern

Nel corso della sprint corrente, sono state apportate piccole estensioni ai pattern Prototype e Simple Factory, già introdotti e descritti nelle iterazioni precedenti. In particolare, per il pattern Prototype sono state aggiunte nuovi ConcretePrototype che rappresentano le forme di Testo e Poligono Irregolare, estendendo così la capacità del sistema di gestire tipologie di oggetti più diversificate e complesse.

Parallelamente, per il pattern Simple Factory sono state integrate le medesime classi come ConcreteProduct, garantendo un'istanziamento coerente e centralizzato degli oggetti.

Infine, sono state apportate correzioni sostanziali all'implementazione del Command Pattern, migliorandone l'aderenza ai principi di progettazione e la chiarezza architetturale, come descritto nella sezione successiva.

4.2.1 Command Pattern: Correzione

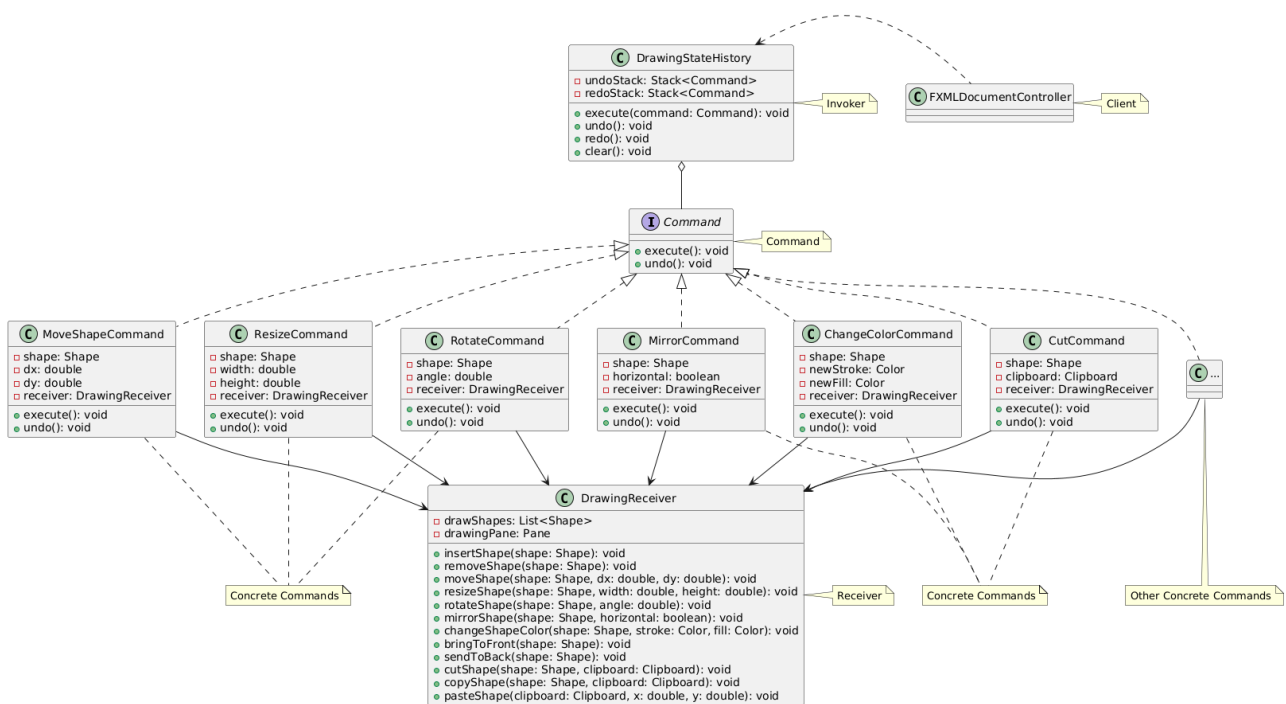


Figura 3: Command Pattern - Correzione

Nota: per semplicità rappresentativa l'immagine presenta solo alcune delle classi che rivestono il ruolo di concrete command

Durante lo svolgimento della terza sprint, il team ha individuato alcune criticità legate all'implementazione del Command Pattern, il cui utilizzo impreciso stava generando una complessità eccessiva nel codice, compromettendo la manutenibilità del sistema e violando alcuni principi fondamentali di progettazione, quali il Single Responsibility Principle e l'Open/Closed Principle. Al fine di prevenire un peggioramento dell'architettura complessiva, si è deciso di destinare un membro del team alla rifattorizzazione del pattern, ritenendo tale intervento prioritario rispetto al completamento di alcuni task residui associati a user story di bassa priorità. Questa scelta ha rappresentato un investimento tecnico strategico, volto a migliorare la qualità del codice e a facilitare le iterazioni future.

Nell'implementazione iniziale, parte della logica operativa risultava distribuita all'interno delle classi concrete che rappresentavano i singoli comandi. Questo approccio, oltre a generare ridondanza, violava il principio di responsabilità singola (*Single Responsibility Principle*), rendendo più complesso il mantenimento e l'estensione del sistema. La correzione ha previsto il trasferimento della logica di esecuzione all'interno della classe *DrawingReceiver*, che ricopre il ruolo di *Receiver*. Tale classe ora centralizza l'effettiva esecuzione delle operazioni (ad esempio, aggiunta, modifica o eliminazione di elementi grafici), mentre i comandi concreti si limitano a delegare tali operazioni al receiver, mantenendo un comportamento uniforme e facilmente gestibile.

Parallelamente, è stato rivisto il ruolo della classe *DrawingHistory*, che funge da *Invoker*. In linea con la struttura canonica del pattern, essa è ora responsabile esclusiva dell'invocazione dei comandi istanziati e della gestione dello storico degli stessi per supportare funzionalità di *undo*, consentendo il ripristino dello stato precedente.

Nel nuovo assetto architetturale, il *FXMLController* assume invece il ruolo di *Client*, ovvero il componente incaricato di creare le istanze dei comandi specifici e di passarle all'*Invoker* per l'esecuzione. Attraverso questa separazione dei ruoli, l'interazione tra l'interfaccia utente e la logica applicativa è stata resa più modulare e facilmente estendibile, secondo i principi della programmazione orientata agli oggetti e del design modulare.

Questa riorganizzazione ha migliorato significativamente la chiarezza architetturale e ha ripristinato una corretta separazione dei ruoli tra le componenti del pattern, favorendo estendibilità e riuso del codice, oltre a semplificare l'implementazione di nuove operazioni grafiche in futuro.

4.3 Third Sprint Backlog

Durante questa sprint, il team ha riscontrato problemi riguardanti la bontà del codice e l'implementazione di alcuni task.

È stato necessario dedicare una parte del tempo di lavoro al refactoring di alcune componenti del sistema. Inoltre, è stata sfruttata pienamente la capacità dei membri di lavorare in parallelo in maniera chiara ed efficace.

Le User Story riguardanti la selezione multipla, il grouping e lo ungrouping (dipendenti dalla selezione multipla stessa) non sono stati completati. Di conseguenza, in quanto previste dallo Sprint Backlog, ma non portate a termine sono state segnate come "NOT DONE". Ciò ha evidenziato una sottostima iniziale del carico di lavoro richiesto, nonché la necessità di un'attenzione costante alla qualità del codice come parte integrante del processo di sviluppo.

Nonostante tali criticità, risultano non completate soltanto le *User Story* a priorità bassa, garantendo così il raggiungimento di una versione del software potenzialmente rilasciabile, aggiornata e coerente con gli obiettivi principali della sprint.

ID	User Story	Story Points	Stato
34	Bug Fix	3	DONE

22	Poligoni irregolari	8	DONE
23	Stringhe	5	DONE
24	Rotazione di una forma	5	DONE
25	Specchiatura	5	DONE
26	Stretch di una forma	5	DONE
27	Selezione multipla	3	NOT DONE
28	Grouping	5	NOT DONE
29	Ungrouping	3	NOT DONE

Tabella 40: Third Sprint Backlog - Report Fine Sprint

4.4 Updated Product Backlog

Backlog ID	User Stories	SP	Backlog ID	User Stories	SP	Backlog ID	User Stories	SP
1	Finestra iniziale	1	13	Modifica della posizione di una forma	2	25	Specchiatura	5
2	Aggiunta forme geometriche	3	14	Modifica del colore di una forma	2	26	Stretch di una forma	5
3	Forme supportate	3	15	Modifica delle dimensioni di una forma	5	27	Selezione Multipla	3
4	Scelta colore del bordo delle forme	3	16	Modifica del livello di una forma	5	28	Grouping	5
5	Scelta colore di riempimento	3	17	Modifica del livello di zoom	5	29	Ungrouping	3

	o delle forme							
6	Salvataggio dei disegni	5	18	Area di disegno espandibile	3	30	Shape creation commands	13
7	Caricamento dei disegni	5	19	Scroll area di disegno	2	31	Memorizzazione dei "Shape creation commands"	8
8	Selezione forma	3	20	Visualizzazione della griglia	3	32	Esportazione di uno "Shapes library file"	8
9	Eliminazione di una forma	1	21	Cambiare dimensioni griglia	3	33	Importazione di uno "Shapes library file"	8
10	Copia e incolla	3	22	Poligoni irregolari	3	34	Bug Fix: selezione dopo spostamento e undo	3
11	Taglia	2	23	Stringhe	5			
12	Undo	5	24	Rotazione di una forma	5			

Tabella 41: Product backlog aggiornato alla terza sprint

4.5 Project Burndown chart:

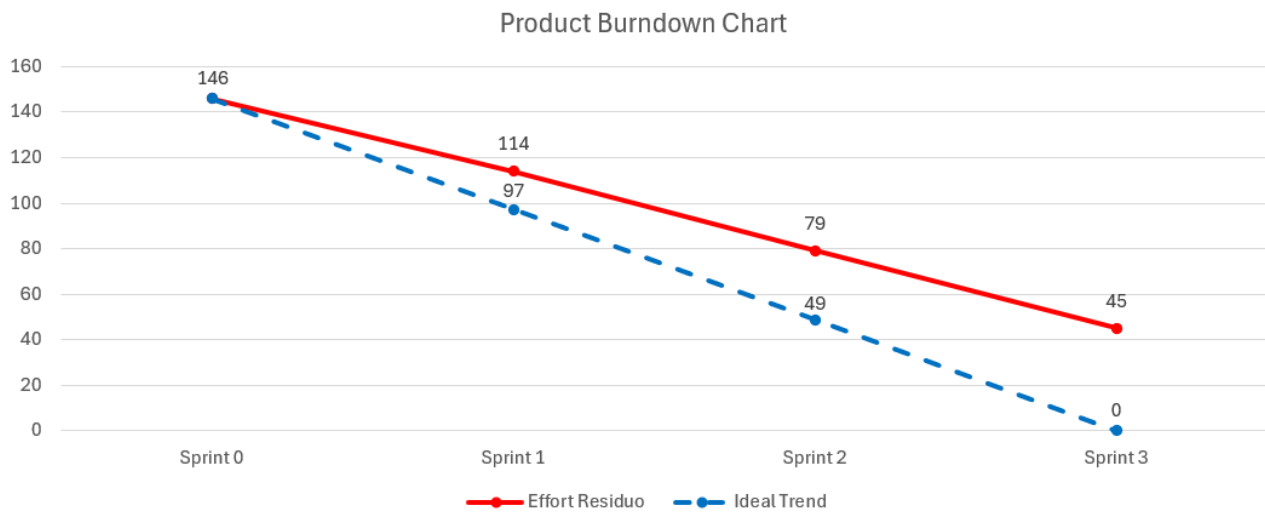


Figura 4.5.1 - Burndown chart (Third Sprint Update)

Il grafico mostra l'andamento del lavoro rimanente (espresso in story points) sprint-by-sprint. La linea tratteggiata blu rappresenta la tendenza ideale di completamento del progetto, mentre la linea rossa mostra l'andamento effettivo.

L'analisi del Burndown Chart della terza sprint evidenzia un divario significativo tra l'andamento ideale stimato e l'effettivo avanzamento del team. In particolare, rispetto alla traiettoria ideale – che prevedeva l'azzeramento degli story points residui (0 SP) al termine dell'ultima iterazione – il team ha concluso la sprint con 45 SP ancora da completare. Anche rispetto alla pianificazione realistica, che stimava una conclusione a 42 SP rimanenti, si registra una leggera deviazione negativa. Tale discrepanza può essere attribuita ai diversi fattori già discussi nelle sezioni precedenti.

In conclusione, il *Burndown Chart* riflette una performance quantitativamente inferiore rispetto al piano, ma qualitativamente giustificabile in virtù delle scelte consapevoli operate dal team per assicurare la qualità del prodotto e la sostenibilità del codice nel lungo termine.

4.6 Third Sprint - Review

4.6.1 Project velocity measured:

Durante la fase di Planning della prima sprint, era stata stimata una velocity pari a 32 story points (SP), successivamente incrementato a 35 SP in occasione del planning della seconda iterazione, in seguito al miglioramento dell'efficienza del team e a una maggiore familiarità con il contesto progettuale. Sulla base di questo trend positivo, il gruppo di lavoro ha deciso di aumentare ulteriormente la stima per la terza sprint, portandola a 42 SP.

Tuttavia, tale stima si è rivelata eccessivamente ottimistica. Durante l'iterazione, infatti, si è reso necessario dedicare una parte significativa del tempo al migliorare la qualità complessiva del sistema e garantire maggiore manutenibilità. Questi interventi non previsti hanno compromesso il completamento dell'intero carico pianificato: a fine sprint, risultavano 11 story points.

Ad ogni modo, la suddivisione iniziale del lavoro è stata equa, infatti, rispetto ai task previsti, ai componenti della squadra sono stati assegnati in media 10.5 story points cadauno.

Nome Membro	Story points
Alessio Bottiglieri	11
Ciro Cutolo	11
Gaetano Frasca	10
Gennaro D'Ambrosio	10

Tabella 42: Suddivisione SP tra i membri del team - 3rd Sprint

Nota: la suddivisione degli story points è stata effettuata rispetto ai task e alla loro importanza nel contesto della storia da cui sono stati determinati.

4.6.2 Problemi riscontrati durante l'implementazione

Diversamente dalle sprint precedenti, la terza è stata conclusa con accumulo di *Technical Debt*. Questo risultato è riconducibile alla produzione iniziale di codice non rispettante i Good Principles della progettazione software, rendendo necessario un ampio refactoring diffuso su molteplici componenti del sistema. Inoltre, alcuni task pianificati hanno richiesto uno sforzo congiunto di più membri, se non l'intero gruppo.

Alcuni componenti del team hanno dovuto abbandonare alcuni dei task per concentrarsi sulle attività di miglioramento del codice.

Data la difficoltà dei vari task, sono stati riscontrati e risolti vari bug in corso di sviluppo, soprattutto per quanto riguarda l'implementazione dei poligoni irregolari e la forma testuale.

4.6.3 Modifiche da apportare al Product Backlog

Durante la seconda release è stato identificato un bug relativo al funzionamento del menù contestuale. Come discusso nelle sezioni precedenti, si è resa necessaria l'inclusione nel Product Backlog di una user story specifica dedicata alla risoluzione di tale difetto. Inoltre, è stata prevista l'esecuzione di test di integrazione funzionali volti a verificare l'effettiva correzione del problema e a garantire la stabilità del sistema.

4.7 Third Sprint - Retrospective

Durante la Sprint Retrospective, il gruppo ha utilizzato lo "Starfish Diagram", come struttura collaborativa per valutare le pratiche del team e identificare le aree di miglioramento continuo.

Durante la Terza Sprint, il team ha fatto consapevolmente riferimento ai diagrammi realizzati durante le interazioni precedenti, usandoli come promemoria delle migliori pratiche concordate e delle insidie comuni da evitare.

Questo approccio si è rivelato particolarmente efficace nel favorire un comportamento coerente ed efficace del team, e nel mantenere un elevato standard di collaborazione, soprattutto nell'area della comunicazione, precedentemente identificata come un'area di crescita.

Il diagramma seguente riflette i progressi del team in diverse aree e conferma che molte delle lezioni dello sprint precedente sono state interiorizzate con successo.

4.7.1 Starfish Diagram

4.7.1.1 Stop (things to stop doing):

- Applicare pattern, senza seguire fedelmente la teoria associata.
- Scrivere codice poco coerente con quello già sviluppato.
- Manipolare codice altrui modificandone la struttura.
- Duplicare metodi già presenti.

4.7.1.2 Less of:

- Lasciare codice commentato nel prototipo.

4.7.1.3 Keep doing:

- Suddivisione equa dei task tra i membri del team.
- Testing incrociato del sistema e delle sue funzionalità.
- Scrivere commenti e documentazione progressiva, durante lo sviluppo.

4.7.1.4 More of (more things to do):

- Rispettare gli orari e l'organizzazione riguardo i daily meetings.
- Comunicazione efficace e periodica.
- Rispettare un formato coerente per i commenti.
- Sfruttare classi di gestione per la logica utile al sistema.

4.7.1.5 Start (things to start doing):

- Lavorare rispettando scrupolosamente i principi fondanti della programmazione "object oriented".

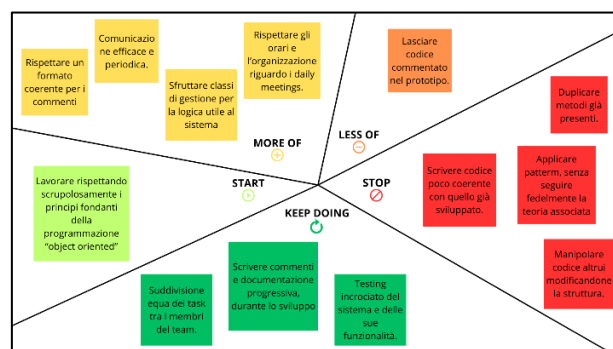


Figura 2: Starfish Diagram - Third Sprint