

SHAPETASTIC - THIRD SPRINT

Alessio Bottiglieri,
Ciro Cutolo,
Gennaro D'Ambrosio,
Gaetano Frasca



Review

Completed stories

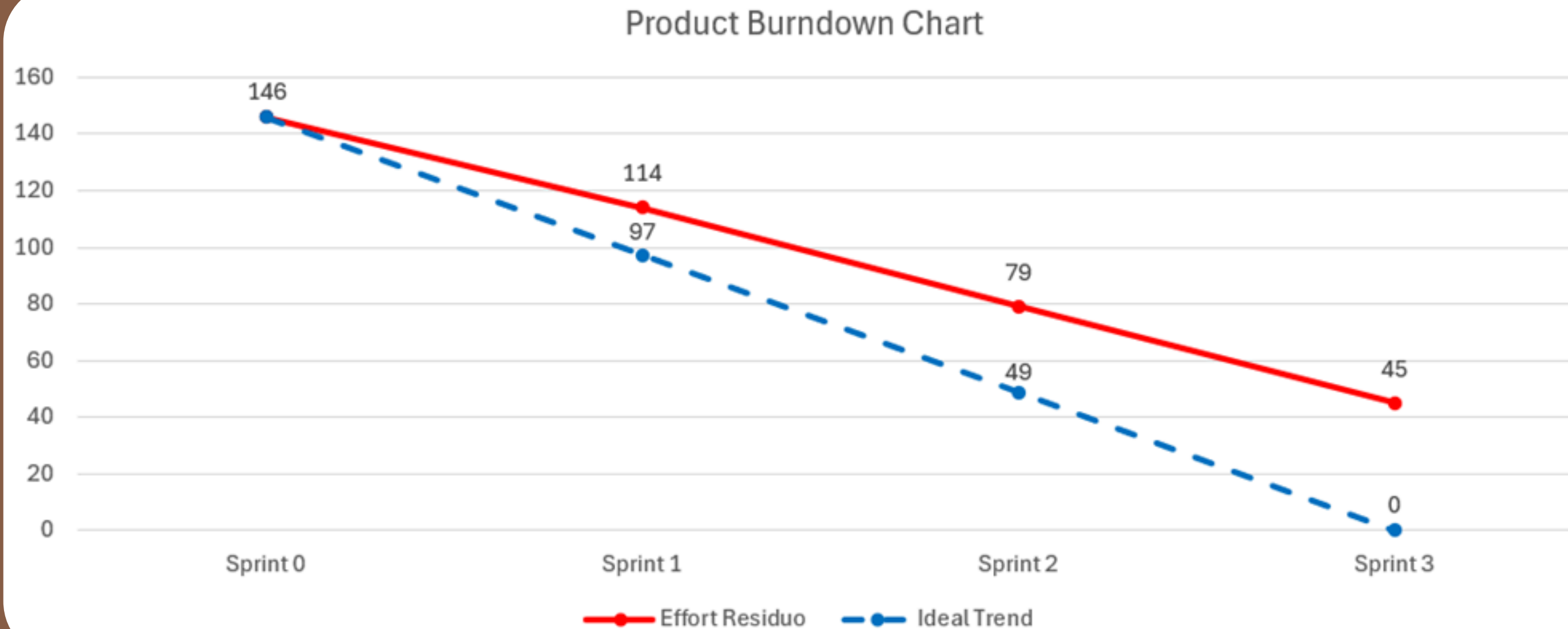
- Storie completate rispetto allo sprint backlog

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	<u>Grouping</u>	5	NOT DONE
29	<u>Ungrouping</u>	3	NOT DONE

Durante questa sprint, il team ha riscontrato problemi riguardanti la bontà del codice e ha riscontrato problemi riguardanti 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.

I task riguardanti la selezione multipla e il grouping/ungrouping (dipendenti dalla selezione stessa) non sono stati completati. Di conseguenza, le user stories previste sono portate a termine in parte, come si può evincere dallo sprint backlog. Questo risultato riflette una sottostima riguardo le storie non completate e la necessità di un maggior focus riguardo la produzione di buon codice.

Burndown Chart



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 terza sprint, il lavoro residuo è di 45 SP. Da questo dato, si può evincere che il team ha sottoperformato rispetto alla previsione ideale (fissa a 0 SP) e a quella realistica (fissa a 34 SP).

Details



Velocity

- Durante la fase di Planning di questa sprint il gruppo di lavoro ha voluto spingersi ancora oltre rispetto alle precedenti sprints, prevedendo 42 SP per la terza sprint.
- Questa stima è risultata esagerata soprattutto perché sono stati necessari aggiustamenti per quanto riguarda la formattazione e la scrittura del codice. Di conseguenza, il team non è riuscito a “bruciare” tutti gli SP previsti, avanzandone 11 in totale.



Issues

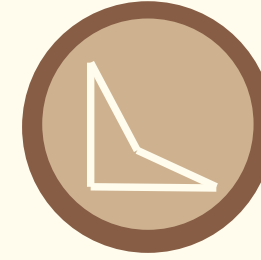
- Diversamente dalle sprint precedenti, la terza è stata completata con accumulo di Technical Debt. Questo risultato è dato dall'iniziale produzione di codice non rispettante i good principles dell'OOP.
- Di conseguenza, alcuni componenti del gruppo hanno dovuto abbandonare alcuni dei task loro assegnati, per effettuare i dovuti miglioramenti al 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.

STARFISH DIAGRAM



Less of

- Lasciare codice commentato, superfluo, nel prototipo da rilasciare.
- Lasciare troppa logica nel controller.



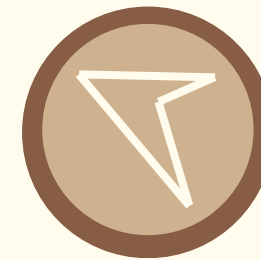
More of

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



Stop doing

- Applicare pattern, senza seguire fedelmente la teoria associata.
- Scrivere codice poco coerente con quello già sviluppato.
- Modificare codice degli altri rendendolo molto diverso da come era in origine.
- Duplicare metodi già presenti.
- Inserire codice non utilizzato da nessun componente.



Start doing

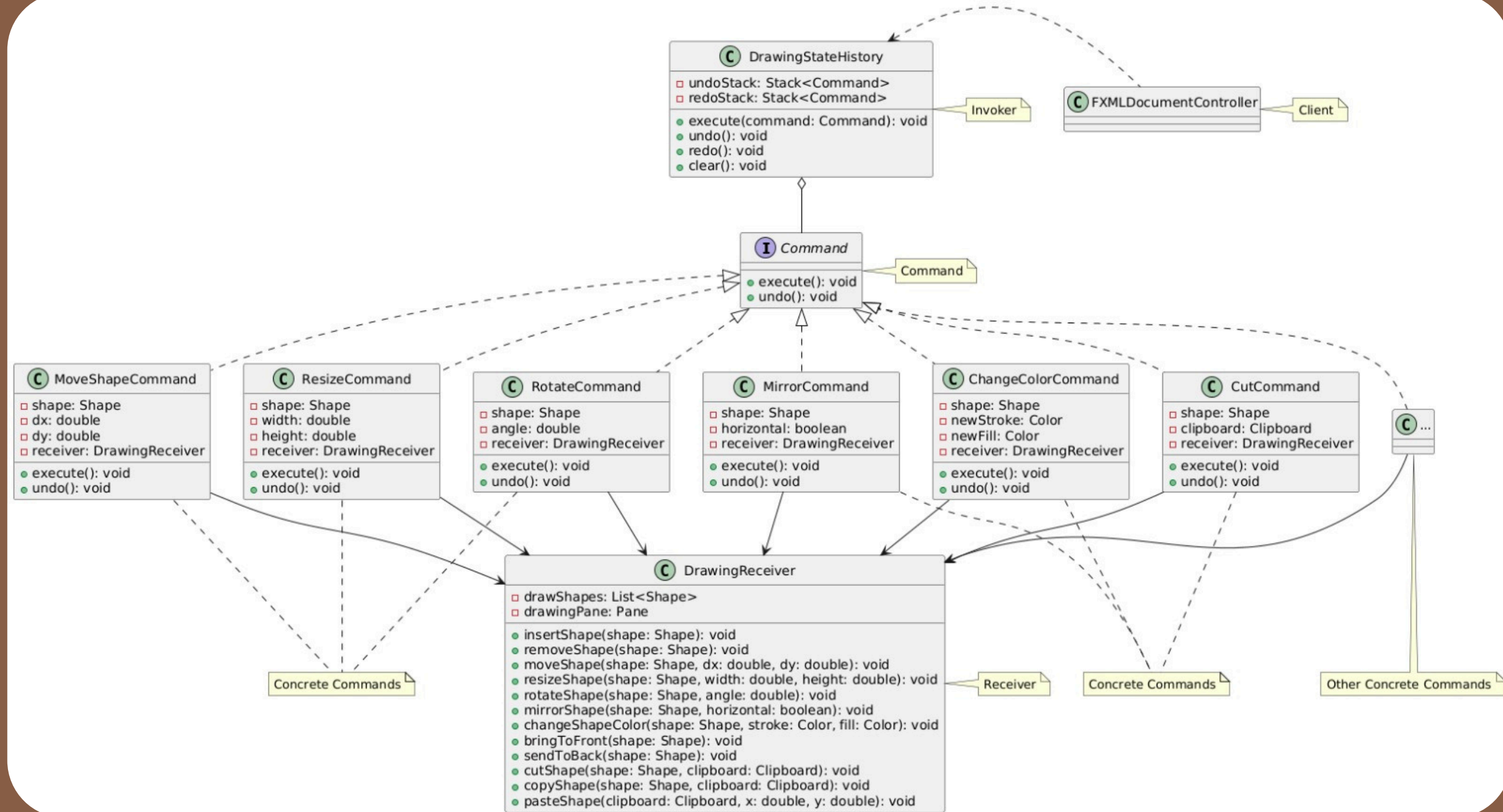
- Segnalare bug e difficoltà riscontrate, anche se risolte.



Keep doing

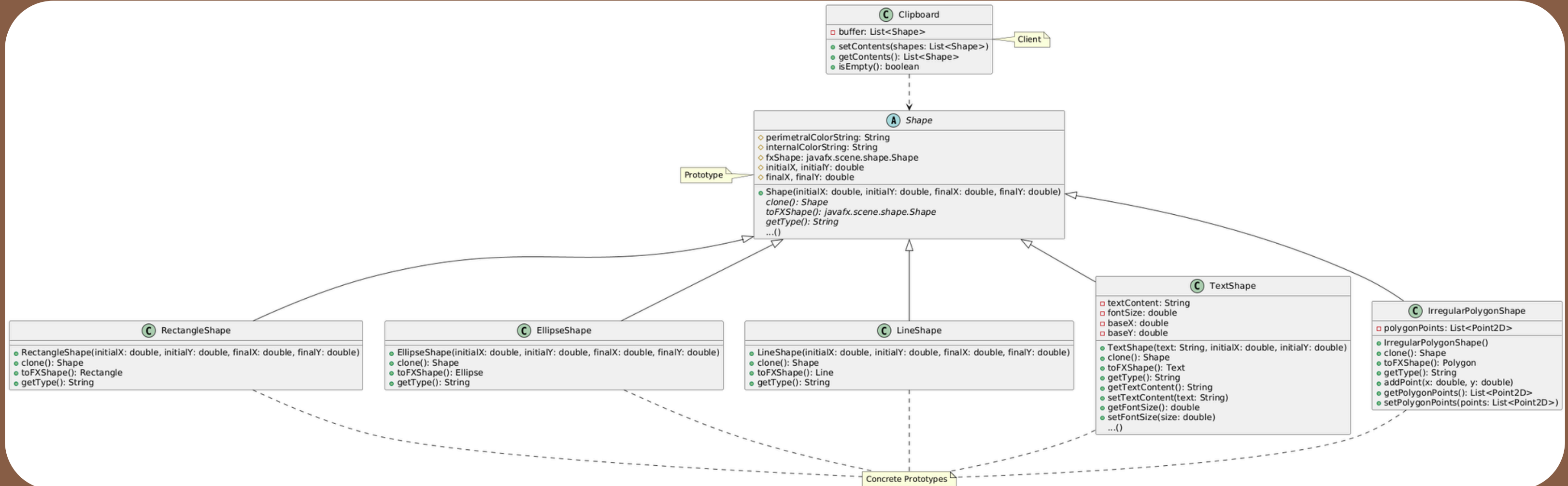
- ·Suddivisione equa dei task tra i membri del team.
- ·Comunicazione sana tra i membri del team.
- ·Lavoro congiunto efficace.
- ·Testing incrociato del sistema e delle sue funzionalità.
- Scrivere commenti e documentazione progressiva, durante lo sviluppo.

COMMAND PATTERN



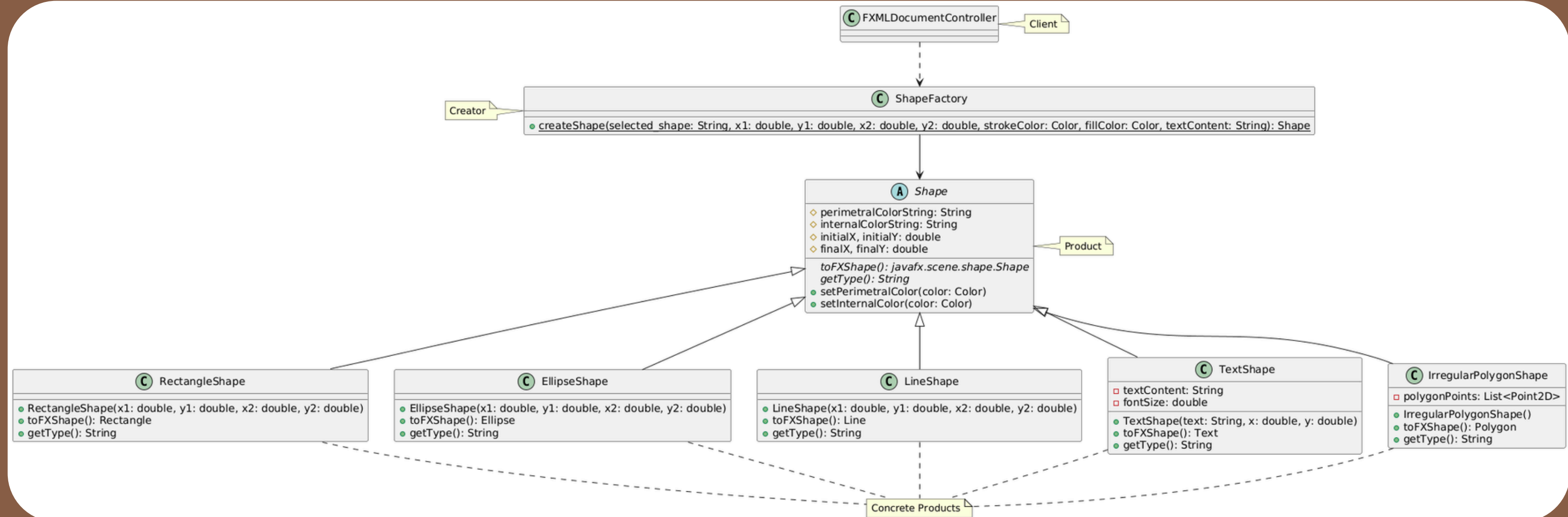
Dato il contesto, vi era la necessità di gestire le operazioni sulle forme in maniera flessibile e annullabile. Il “Command Pattern” permette proprio di incapsulare le operazioni in diversi oggetti che implementano un’interfaccia comune. Questa soluzione permette di separare la richiesta di azione dalla sua concreta esecuzione e permette di gestire facilmente l’annullamento dell’azione stessa.

PROTOTYPE PATTERN



Nel contesto attuale, è richiesto un elevato tasso di creazioni delle varie forme, tra preview e quant'altro. Applicare il “Prototype Pattern” permette la creazione efficiente di nuovi oggetti “Shape” concreti, clonando istanze esistenti ed evitando i costi elevati legati alla costruzione da zero. Inoltre, questo garantisce rapidità di creazione, coerenza nella gestione e indipendenza tra le copie.

FACTORY PATTERN



Per lo sviluppo di questa applicazione, la creazione di oggetti di tipo `Shape` rischiava di diventare dispersiva, soprattutto con l'introduzione di più forme sfruttanti i medesimi parametri. Di conseguenza, è stato utilizzato il "Simple Factory Pattern" in modo da centralizzare la creazione delle forme e ridurre la duplicazione del codice, nonchè separare creazione e utilizzo e migliorare manutenibilità e riusabilità.