

Team ShiftsHappen

Basili Matteo, Buniy Massimo, Cappellini Federico, Finocchi Alessandro

MS3 – Medical Staff Shift Scheduler



AGENDA



1 Introduzione

2 Sprint di progetto

A Sprint 0

D Sprint 3

B Sprint 1

E Sprint 4

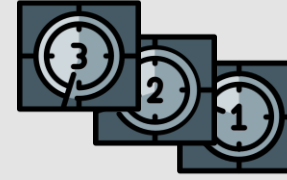
C Sprint 2

F Sprint 5

2 Analisi della produzione

3 Ai posteri

Introduzione

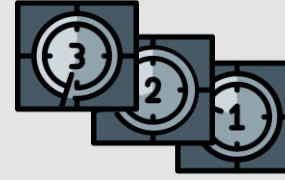


Nell'anno accademico 2024/2025 il progetto MS3 ha avuto un'evoluzione in direzione dei seguenti aspetti:

- **Autorizzazione e Autenticazione**
- **Multi-tenancy**
- **Analisi statica e dinamica**
- **Reintegrazione dei test**
- **Documentazione**



Introduzione

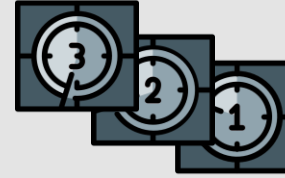


Altre attività portate avanti hanno riguardato:

- **Omogeneizzazione della grafica**
- **Continuo traduzione interfacce**
- **Aggiornamento design di alcune schermate**
- **Aggiunta nuove funzionalità**

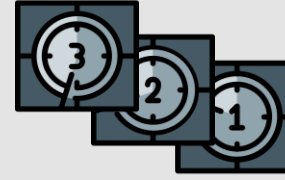


Obiettivi



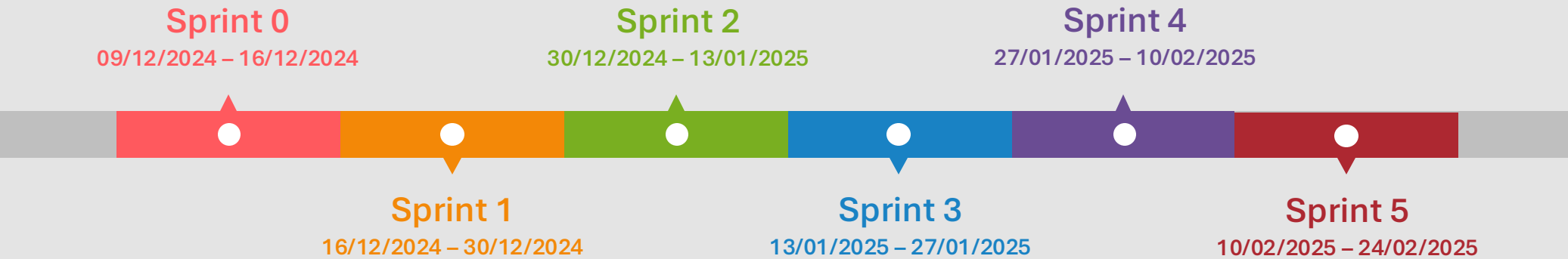
- Realizzare dei meccanismi di sicurezza, in particolare riguardo l'**autenticazione** e l'**autorizzazione**, ragionando nell'ottica del multi-tenancy
- Integrare strumenti di analisi **statica** e **dinamica** nel flusso di sviluppo del codice
- Reintegrare i **test** nel ciclo di realizzazione del sistema

Nuove funzionalità



- Creazione di turni per i servizi medici da parte del **Configuratore**
- Cancellazione di turni per i servizi medici da parte del **Configuratore**
- Visualizzazione dei turni incompleti da parte del **Pianificatore**

Sprint Timeline



In totale sono stati fatti **6 sprint**:

- **5 + uno** iniziale, lo *sprint 0*
- Durata media: **2 settimane**.



Sprint 0

09/12/2024 – 16/12/2024

Sprint 0



- **Durata:** 1 settimana
- **Periodo:** 09/12/2024 – 16/12/2024
- **Attività principali:**
 - Esecuzione del progetto sia in **locale** che su **Docker**
 - Studio del progetto, analisi della documentazione e organizzazione del lavoro di team
 - Completamento dei **task** assegnati per questo sprint



Sprint 0



- **Esito:** lo sprint è andato a buon fine
 - Tutti i membri del team sono riusciti a eseguire il progetto sia in locale che su Docker
 - Per migliorare la nostra conoscenza del progetto, abbiamo:
 - Studiato la documentazione delle versioni precedenti
 - Strutturato un sistema di **versioning** per organizzare il codice in modo efficace
 - Analizzato l'architettura del progetto, comprendendo la sua configurazione e la struttura dei package
 - Abbiamo lavorato per chiudere i task assegnati

Sprint 0



- **Problemi riscontrati:**
 - **Alto debito tecnico**, che ha reso alcune parti del codice più difficili da comprendere e modificare
 - Difficoltà nel chiarire alcuni termini concettuali, come la differenza tra "*Specializzazione*" e "*Medical Service*"
 - Issue relativa ad **autorizzazione e autenticazione**
 - Dalla sezione modifica profilo, cambiando id nell'url posso accedere e modificare i profili degli altri utenti
 - ❑ [GitHub issues #282](#)
 - Errori sulla console del browser durante l'esecuzione dell'applicazione

Sprint 0



Task

✓	2	Mettere testo con mail e ruolo nella schermata di login
✓	1	Nella pagina utenti rinominare la colonna attore con ruolo
✓	2	Nella pagina utente evidenziare un pulsante nell'header per riordinare le colonne
✓	2	Nella pagina configurazione vincoli sistemare la grafica dei componenti
✓	1	Modificare il copyright
✓	1	Modificare il messaggio nel login per rendere più generale il messaggio di fallimento di login
✓	5	Nella pagina servizi la modifica del servizio deve apparire un menù a tendina che lo suggerisce
✓	5	Nella pagina servizi mettere in italiano la mansione (attenzione perchè questo ha effetto sul DB, c'è del debito tecnico)
✓	2	Nella pagina di gestione festività e in calendario il footer sta in mezzo allo schermo
✓	3	Sezione modifica profilo dummy
✓	5	Nel profilo, le specializzazioni sono elencate, dovrebbero essere ricercabili tramite una barra di testo che suggerisce la specializzazione nel mentre che la si cerca
✓	1	Nel profilo cambiare lastname con Cognome
✓	1	Nei parametri di configurazione dei vincoli controllare che i caratteri siano numerici (interi) e maggiori di zero
✗	2	Il configuratore non può assegnare una guardia o modificare i turni
✗	8	Coinvolgere anche oncologia in una schedulazione
✗	13	Nella pagina configurazione vincoli testare che il cambiamento della configurazione abbia effetto nello scheduling
✗	13	Testare rigenera pianificazione

Tot done:

31



Sprint 1

16/12/2024 – 30/12/2024

Sprint 1



- **Durata:** 2 settimane
- **Periodo:** 16/12/2024 – 30/12/2024
- **Attività principali:**
 - Introduzione della **sicurezza** tramite **RBAC**¹ con **Spring Security** all'interno del progetto
 - Studio del **multi-tenancy** e ricerca di soluzioni compliant
 - Completamento di alcuni **task** mancati durante lo sprint precedente



1. Il pattern di sicurezza Role Based Access Control

Sprint 1



JWT

```
{  
  [...]  
  "jwt":  
    "eyJhbGciOiJIUzI1NiJ9.eyJyYb2xliJpbIIJPTEVfRE9DVE9SliwiUk9MR  
    V9QTEFOTkVSII0sInN1YiI6Imdpb3Zhbm5pY2FudG9uZUBnbWFPb  
    C5jb20iLCJpYXQiOiE3MzUzNDIyNjE3MzUzNDIyNjE3MzUzNDIyNjE3  
    X0.dqFaMH7Ws2ZK0q00_sLcUXgDEczb_vSLPLBGEs5f8YU"  
}
```



```
{  
  "alg": "HS256"  
}
```



```
{  
  "role": [  
    "ROLE_DOCTOR",  
    "ROLE_PLANNER"  
  ],  
  "sub": "giovannicantone@gmail.com",  
  [...]  
}
```

Sprint 1



Mappa API-Utente

#	API Endpoint	Metodo	Dottore	Pianificatore	Configuratore
59	/api/notification/id={userId}	GET	✓	✓	✓
60	/api/notification/updateStatus	PUT	✓	✓	✓
61	/api/concrete-shifts/	GET	✓	✓	✓
62	/api/concrete-shifts/	PUT	✗	✓	✗
63	/api/concrete-shifts/	POST	✗	✓	✗
64	/api/concrete-shifts/available-users-for-replacement/	POST	✓	✗	✗
65	/api/concrete-shifts/user_id={userId}	GET	✓	✗	✗
66	/api/concrete-shifts/{idAssegnazione}	DELETE	✗	✓	✗
67	/api/conditions	GET	✓	✓	✓

✓ Accesso consentito

✗ Accesso negato

Sprint 1



Autorizzazioni a livello del singolo metodo

```
@PreAuthorize("hasAnyRole('CONFIGURATOR')")  👤 massimo
@RequestMapping(method = RequestMethod.POST, path = "/updateService")
public ResponseEntity<?> updateService(@RequestBody MedicalServiceDTO service) {
    if (service != null) {
```

```
@PreAuthorize("hasAnyRole('CONFIGURATOR', 'DOCTOR', 'PLANNER')")  👤 simone
@RequestMapping(method = RequestMethod.GET)  🌐
public ResponseEntity<?> getAllDoctors() {
    Set<MedicalDoctorInfoDTO> medicalDoctorInfoDTOSet = doctorController
```

Sprint 1



Rimozione la selezione del ruolo nella fase di login

Login

Indirizzo Email

Password

Accedi

☐ Ricordami

[Password Dimenticata?](#)

Development shortcut

Seleziona ruolo da assegnare



Accedi come:
Configuratore



Accedi come:
Dottore



Accedi come:
Pianificatore

Seniority	
mail.com	Specialista Junior <input type="button" value="Insert"/>
gmail.com	Specialista Senior <input type="button" value="Insert"/>
gmail.com	Strutturato <input type="button" value="Insert"/>
mail.com	Specialista Senior <input type="button" value="Insert"/>
gmail.com	Specialista Senior <input type="button" value="Insert"/>
Dottore, Configuratore, Planner	
fullpermessi@gmail.com	Strutturato <input type="button" value="Insert"/>



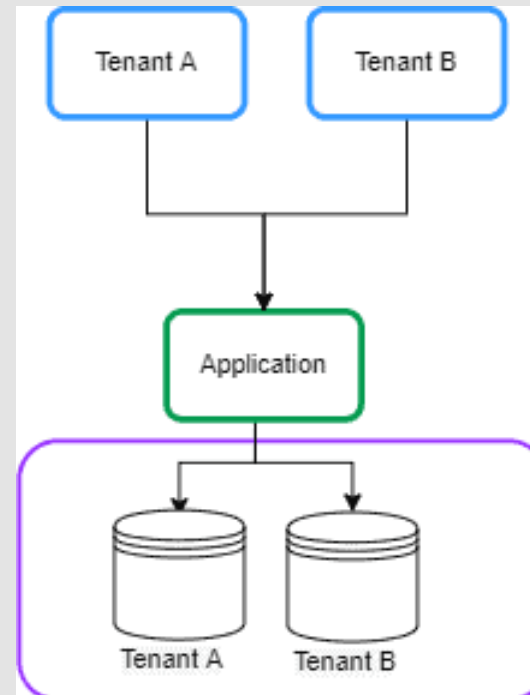
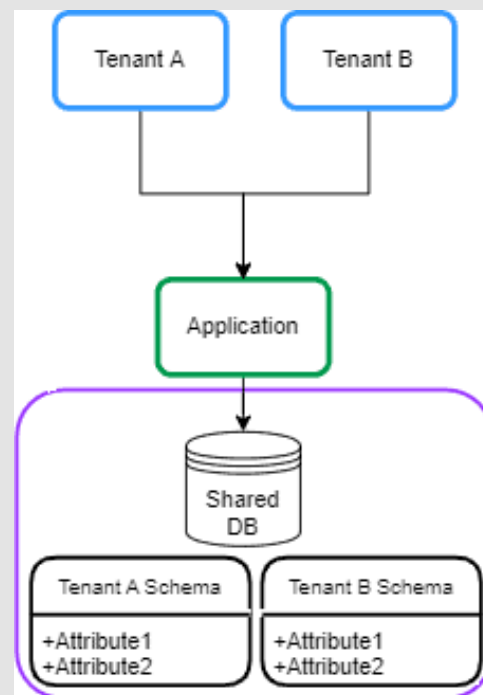
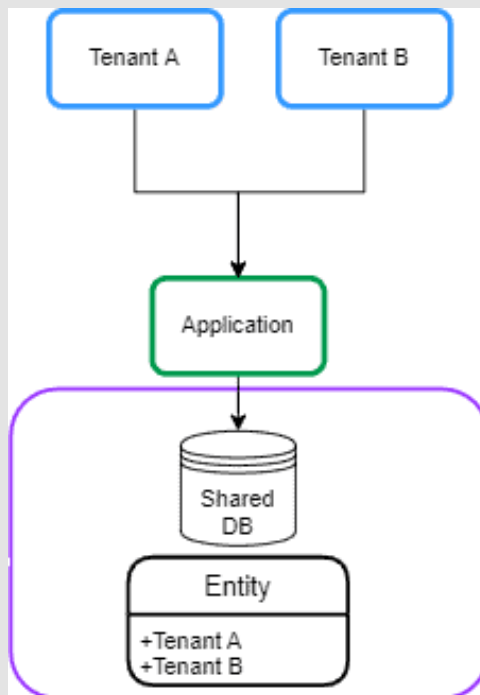
Multi-tenancy: Memorizzazione e determinazione del tenant ID

Subdomain-based MT Pattern: <code>https://<tenant>.yourdomain.com</code> Pros: la separazione è pulita e permette un branding del tenant perchè l'URL è customizzato per ognuno Cons: Serve un supporto per il routing/DNS dei sottodomini	Path-based MT Pattern: <code>https://yourdomain.com/<tenant>/...</code> Pros: Più semplice del subdomain-based, basta un filtro che parsi l'header e un unico dominio Cons: L'URL può diventare complesso e ci possono essere problemi con il caching delle richieste HTTP
Header-based MT Pattern: Custom header Pros: L'URL rimane pulito ed è semplice da implementare a livello HTTP Cons: L'header deve essere sanitizzato per evitare vulnerabilità di sicurezza	JWT-claim-based Pattern: Store the tenant ID in a JWT claim Soluzione proposta: Il token incapsula il tenant in modo sicuro e compatto essendo cifrato e firmato, oltre a far rimanere l'URL pulito. Inoltre, si integra bene con il framework di sicurezza che stiamo utilizzando.

Sprint 1



Multi-tenancy: Partizionamento dei dati



Sprint 1



Task completati

- Problema del **footer** definitivamente risolto
 - In alcune pagine si trovava in mezzo allo schermo
- Migliorata la coerenza grafica dell'applicazione
 - I **drawer** che appaiono dal basso adesso coprono sia il footer che la **dashboard** laterale
 - Tutte le pagine principali hanno lo stile di una **card**
- Negata al configuratore la possibilità di modificare i turni di lavoro assegnando o meno una guardia

Sprint 1



Task completati

- Testato che la configurazione dei vincoli avesse effetto nello scheduling e si è notato che:
 - Il vincolo *Massimo numero di ore consecutive* si riferisce ad un turno, non ad una persona, a differenza degli altri vincoli
 - Assegnare un medico ad un turno ha lo stesso peso in **Uffa Points** che sceglierlo come disponibile
 - Se non ci sono abbastanza medici, lo scheduling di un turno viene prodotto parzialmente

Sprint 1



- **Esito:** lo sprint è andato a buon fine
 - Integrato Spring Security come framework di sicurezza
 - Implementata autenticazione basata su **JWT**, migliorando la gestione degli utenti e la protezione dei dati
 - Introdotte restrizioni di accesso alle **API**, limitando l'uso solo agli utenti autorizzati
 - Controllo basato sui ruoli (RBAC), in cui i permessi vengono assegnati in base ai ruoli degli utenti
 - Cifratura delle password con **BCrypt**
 - Esplorazione delle soluzioni di Multi-tenancy più adatte un sistema scalabile e capace di gestire clienti indipendenti

Sprint 1



- **Problemi riscontrati:**
 - Vulnerabilità ad attacchi **XSS (Cross-Site Scripting)**
 - Necessità di inserimento di JWT nei **cookie HTTP-Only** ed utilizzo di **HTTPS**
 - API mai utilizzate o non ancora implementate
 - Difficoltà nel comprendere il loro funzionamento
 - Difficoltà nel mapping di autorizzazione **API-Utente**
 - Il **diagramma E-R** non aggiornato¹
 - Difficoltà nel comprendere il funzionamento dell'applicazione
 - Allungamento nello svolgimento dei relativi task
 - Molti **test** non più validi a causa di precedenti refactoring
 - Impossibilità di riutilizzo, necessità di attenzione

1. Il meccanismo di snapshot degli uffa point non è documentato all'infuori del precedente sprint 7 in cui però non è spiegato

Sprint 1



Task

Stato	Pesi	Descrizione
✓	3	Nella pagina di gestione festività e in calendario il footer sta in mezzo allo schermo
✓	1	Nei parametri di configurazione dei vincoli controllare che i caratteri siano numerici (interi) e maggiori di zero
✓	2	Il configuratore non può assegnare una guardia o modificare i turni
✓	8	Coinvolgere anche oncologia in una schedulazione
✓	13	Nella pagina configurazione vincoli testare che il cambiamento della configurazione abbia effetto nello scheduling
✓	3	Scrivere documento per specificare quale API è autorizzato a chi. Deve essere ragionato nell'ottica del multi-tenancy, (versione 0.1: ToC e mappa api-utente)
✓	5	Presentare le opzioni di multi-tenancy adottabili specifiche a questo progetto (schema architetturale)
✓	5	Spring security all'interno del progetto (storia tecnica: inserirlo, e farlo funzionare per esempio con bottoni dummy che possono essere usati solo da chi ha un ruolo specifico)

Tot to review:	40
Tot:	40

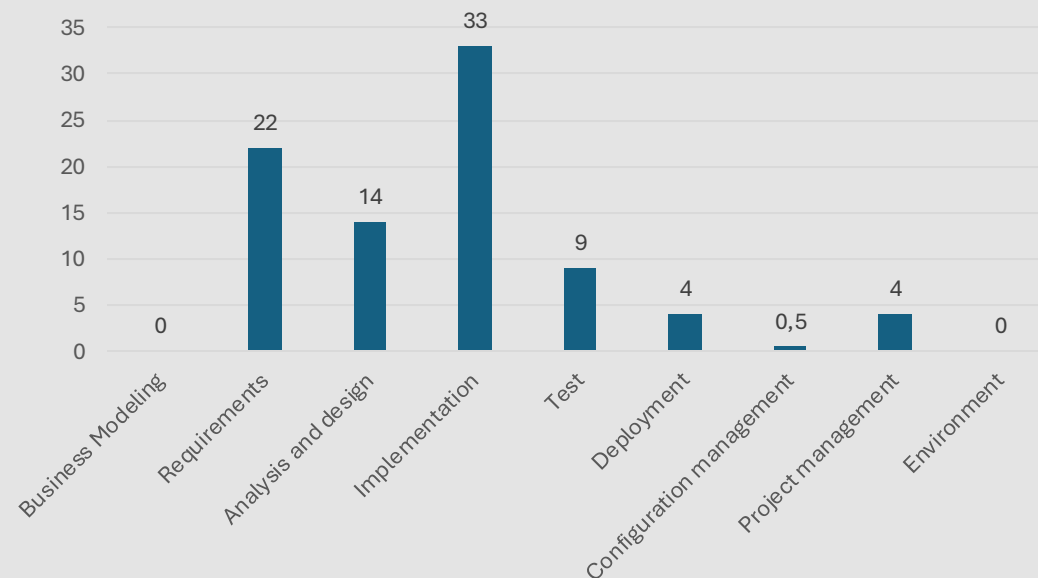
INIZIO:	16/12/24
FINE:	30/12/24

Sprint 1



Analisi per disciplina *RUP*

	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili		2					20,5	1		23,5	26,55%
Massimo Buniy			4	23				1		28	31,64%
Federico Cappellini						4				4	4,52%
Alessandro Finocchi		2	6	14	9			2		33	37,29%
Totale per disciplina	0	4	10	37	9	4	20,5	4	0	88,5	
Percentuale per disciplina	0,00%	4,52%	11,30%	41,81%	10,17%	4,52%	23,16%	4,52%	0,00%		



Sprint 1



Conclusioni

- Pone fine alla fase di "esplorazione" del sistema esistente
- Segna l'inizio dell'attenzione verso gli aspetti di sicurezza e di multi-tenancy all'interno del progetto
- Produce un'importante documento riguardo *l'API authorization*
 - Può essere consegnato al cliente, il quale non deve far altro che compilarlo con le giuste autorizzazioni



Sprint 2

30/12/2024 – 13/01/2025

Sprint 2



- **Durata:** 2 settimane
- **Periodo:** 30/12/2024 – 13/01/2025
- **Attività principali:**
 - Test di integrazione
 - Sviluppo delle due soluzioni di **multi-tenancy** su delle **mini applicazioni** ed integrazione di entrambe le soluzioni su due **branch** del progetto
 - Studio di soluzioni di analisi del codice
 - Creazione dei turni da interfaccia grafica



Sprint 2



Multitenancy

Come il backend determina il tenant ID per ogni richiesta

JWT-claim-based

Pattern: salva il tenant ID in un claim JWT

Soluzione: Il token incapsula il tenant in modo sicuro e compatto essendo cifrato e firmato, oltre a far rimanere l'URL pulito. Inoltre, si integra bene con il framework di sicurezza che stiamo utilizzando.

```
public String generateToken(CustomUserDetails userDetails) {  
    Map<String, Object> claims = new HashMap<>();  
    claims.put("role", userDetails.getAuthorities().stream()  
        .map(GrantedAuthority::getAuthority)  
        .collect(Collectors.toList()));  
    claims.put("current_tenant", userDetails.getTenant().toLowerCase());  
  
    return createToken(claims, userDetails.getUsername());  
}
```

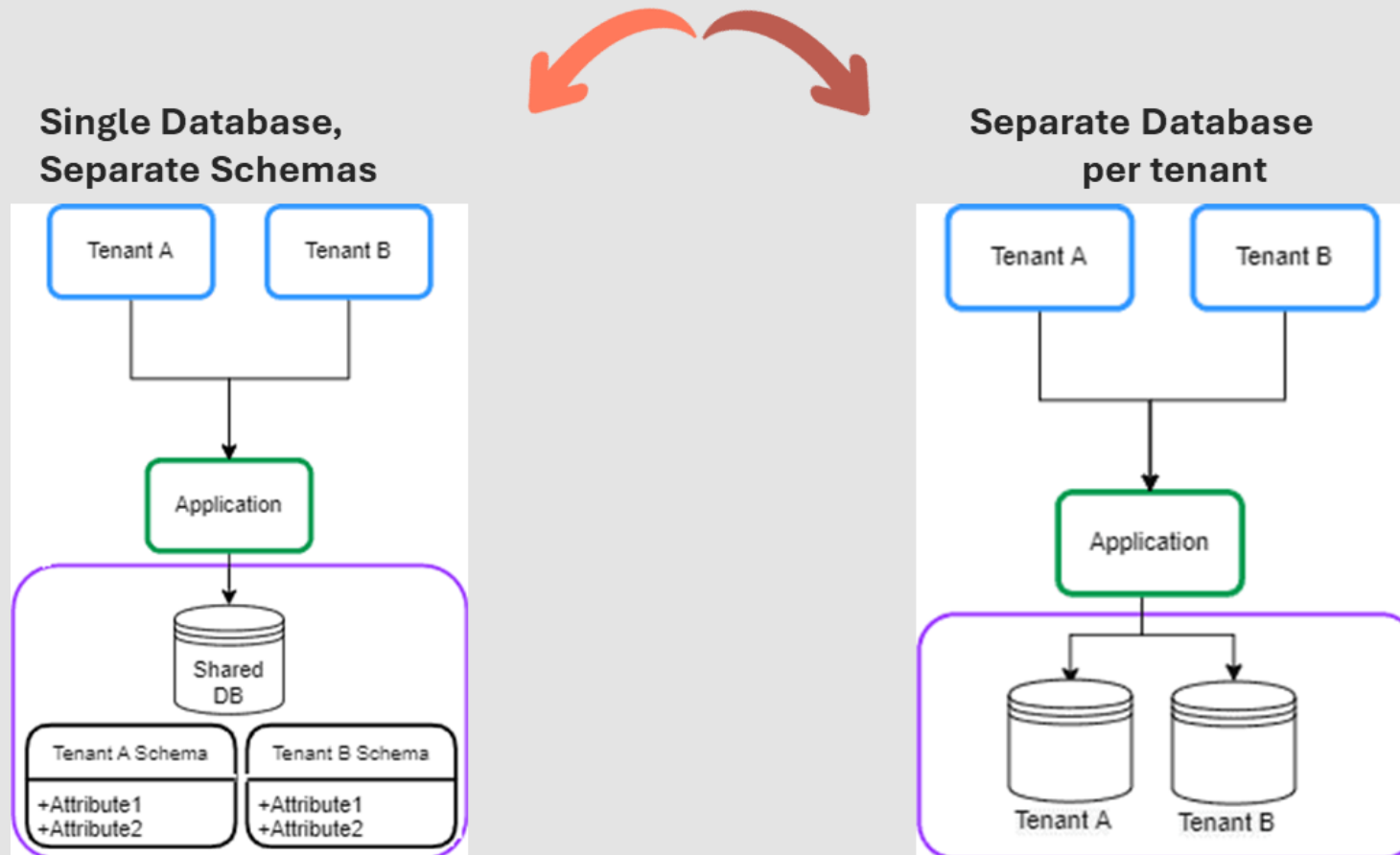


Sprint 2



Multitenancy

Partizionamento dei dati



Sprint 2



Analisi statica

Strumenti Backend (Java)

SonarCloud: analisi approfondita della qualità del codice, rileva bug, vulnerabilità e code smells

SpotBugs: Strumento leggero e specializzato per rilevare bug specifici del codice Java. Facilmente integrabile in IntelliJ tramite Plugin.

PMD: Completa SpotBugs rilevando problemi di stile e migliorando la leggibilità del codice.

Strumenti Frontend (React)

ESLint: Standard de facto per l'analisi del codice JavaScript. Rileva problemi di sintassi e aderenza alle best practice.

Lighthouse: Strumento indispensabile per analizzare performance, SEO, e accessibilità delle applicazioni web.

DependaBot: Strumento di GitHub che automatizza la gestione delle dipendenze nei progetti software.

Sprint 2



Analisi dinamica

Java Profilers

Goal: fornire insight su vari aspetti di runtime di un applicazione (memory usage, cpu consumption, garbage collection, potential bottlenecks...)

VisualVM: tool open-source free che fa profiling di cpu in real-time, memoria, thread, GC, oltre a permettere anche analisi di thread

Jprofiler, YourKit: commerciali

Sec.&Vulnerability Scanners

OWASP ZAP: tool open-source free che prova a sfruttare vulnerabilità conosciute (SQL injection, XSS, ...). E' facile da automatizzare in pipeline di CI/CD e ha un ecosistema di plugin molto vasto per ampliare le sue features.

Necessità di esperienza nell'ambito della sicurezza per interpretare correttamente i risultati.

Burp suite: commerciale, più features

Monitoring&Observability

Spring Boot Actuator: built-in dentro Spring, fornisce diverse features per aiutare a monitorare una app spring con la sua health, environment, dump, e si integra facilmente con altri sistemi come Prometheus.

Utilizza gli endpoint HTTP per interagire con l'applicazione.

New relic, Datadog: commerciali

Frontend analysis

React Dev Tools: estensione di browser per debugging di componenti React, permette di ispezionarne l'architettura, i cambi di stato e le performance al rendering. Fatto apposta per un'architettura React component-based.

Così si possono trovare bottleneck nel rendering e inefficienze nella gestione dello stato

Sentry: commerciale

Sprint 2



UI per inserimento dei turni durante la creazione di un servizio

The image displays three overlapping UI mockups illustrating the process of creating a service and adding shifts.

Top Mockup: Crea un nuovo servizio

- Header: CLINIC, EMERGENCY, OPERATING ROOM, WARD
- Form: Crea un nuovo servizio
- Input: Nome del Servizio (dropdown)
- Section: Seleziona le mansioni da assegnare:
- Options: ☐ Clinica, ☐ Pronto Soccorso, ☐ Ambulatorio, ☐ Sala Operatoria

Middle Mockup: Clinica

- Header: Clinica
- Section: Aggiungi un nuovo turno per la mansione Clinica
- Form: Crea un nuovo turno
- Fields: Fascia oraria (radio buttons: Mattina, Pomeriggio, Notte), Orario di inizio del turno (08:00), Durata del turno (06:00)
- Fields: Medico strutturato (1), Medico specializzando junior (1), Medico specializzando senior (1)
- Section: Giorni della settimana:
- Options: ☐ Lunedì, ☐ Martedì, ☐ Mercoledì, ☐ Giovedì, ☐ Venerdì, ☐ Sabato, ☐ Domenica
- Buttons: CANCELLA, AGGIUNGI TURNO

Right Mockup: Clinica

- Header: Clinica
- Section: Dettagli turno
- Fields: Fascia oraria: Mattina, Giorni della settimana: Lunedì, Ora di inizio: 08:00, Durata del turno: 6 ore 0 minuti
- Buttons: < INDIETRO, CREA

Sprint 2



Task completati

- Implementati controlli per verificare che ad ogni mansione di ogni turno vi sia almeno un medico strutturato
- Possibilità di creazione di uno schedulo di un giorno
- Risolto il problema dello stop della schedulazione al primo vincolo non soddisfatto
 - ora i turni verranno riempiti il più possibile
 - un turno è infattibile se per definizione nessun medico strutturato è disponibile per almeno una delle mansioni

Sprint 2



- **Esito:** lo sprint è andato a buon fine.
 - Metà dei package di **test** sono stati reintegrati
 - Introdotte le due soluzioni di **multi-tenancy** all'interno di due branch del progetto
 - Prodotti due documenti sullo studio dei tool di **analisi statica e dinamica** per il progetto
 - Creata un interfaccia grafica più complessa per la creazione di un servizio, permettendo di definire anche la **creazione** dei relativi **turni**

Sprint 2



- **Esito:** lo sprint è andato a buon fine.
 - Implementati controlli sul fatto che ad ogni mansione di ogni turno vi fosse almeno un medico strutturato
 - Aggiunta la possibilità di creare uno scheduling di un giorno
 - Risolta la generazione di una schedulazione che si fermava al primo vincolo non soddisfatto

Sprint 2



- **Problemi riscontrati:**
 - Non effettiva separazione dei dati tra i vari tenant
 - Le operazioni database vengono fatte da un unico utente, il quale ha i permessi di accesso a tutti i dati di tutti i tenant

Sprint 2



Task

Stato	Pesi	Descrizione
✓	5	Risolvere il problema riscontrato nella schedulazione
✓	3	Test di integrazione: ripristino test già presenti
✓	4	Studiare le soluzioni di analisi statica (e poi dinamica se non ha richiesto troppo tempo quella statica) del codice
✓		Sviluppo MT soluzione 2
✓		Sviluppo MT soluzione 3
✓		Integrare soluzioni MT nel progetto
✓	5	Aggiungere la possibilità di creare tumi ad un servizio medico

Tot done:	17
Tot:	17

INIZIO:	30/12/24
FINE:	13/1/25

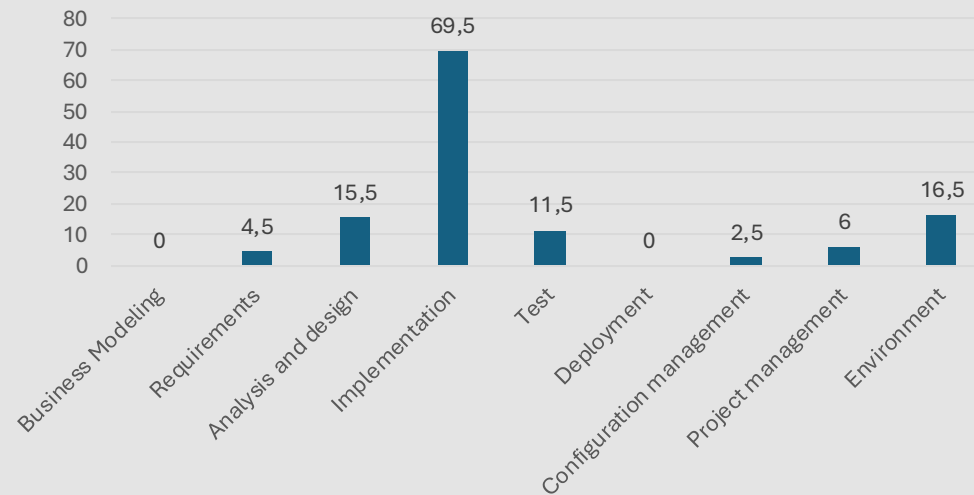
Nota: non sono stati definiti i pesi di ogni task in questo sprint

Sprint 2



Analisi per disciplina *RUP*

	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili			7	22			9	4		42	32,68%
Massimo Buniy		2	1,5	24,5						28	21,79%
Federico Cappellini			2	30			2,5			34,5	26,85%
Alessandro Finocchi		2,5	4	2	13			3,5		25	18,68%
Totale per disciplina	0	4,5	14,5	78,5	13	0	11,5	7,5	0		128,5
Percentuale per disciplina	0,00%	3,50%	11,28%	60,31%	10,12%	0,00%	8,95%	5,84%	0,00%		



A photograph of a red running track with white lane lines. Several starting blocks are visible, arranged in a diagonal line across the frame. The blocks are silver with red rubber footplates. The text "Sprint 3" is overlaid in white on the middle of the image.

Sprint 3

13/01/2025 – 27/01/2025

Sprint 3



- **Durata:** 2 settimane
- **Periodo:** 13/01/2025 – 27/01/2025
- **Attività principali:**
 - Segregazione dei dati nelle due soluzioni¹ di multi-tenancy
 - Integrazione dei tool di analisi del codice
 - Implementazione soft delete
 - Continuare il ripristino dei test
 - Completare task relative alla gestione dei turni



Sprint 3



Single Database, Separate Schemas

Ogni tenant ha il proprio schema nello stesso database fisico

- Soluzione:

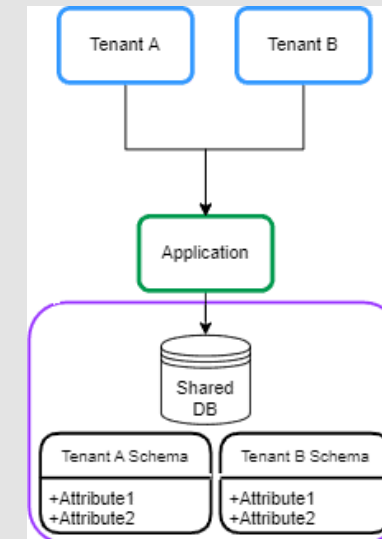
- Spring/Hibernate può passare dinamicamente da uno schema all'altro in base al tenant, determinato durante il runtime
- Ogni tenant ha un utente con privilegi limitati sul suo schema

- Vantaggi:

- Migliore segregazione dei dati
- Più semplice eliminare o migrare un singolo tenant

- Svantaggi:

- Il carico alto di un tenant può influenzare gli altri tenant
- Gestione potenzialmente complicata di molti schemi in un unico database



Sprint 3



Implementazione (Single Database, Separate Schemas)

- Connessioni dinamiche personalizzate:

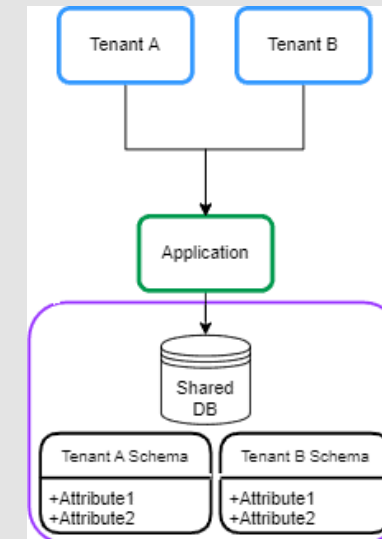
- La classe *DataSourceConfig* crea connessioni con credenziali specifiche per ogni tenant
- Ogni tenant dispone di un utente configurato con privilegi limitati sul relativo schema

- Gestione dinamica dei tenant:

- La classe *SchemaSwitchingConnectionProviderPostgreSQL* imposta il contesto SQL per isolare le query al solo schema del tenant

- Integrazione con Hibernate:

- Hibernate utilizza un *MultiTenantConnectionProvider* per gestire le connessioni multiple e un *CurrentTenantIdentifierResolver* per risolvere dinamicamente l'identificatore del tenant



Sprint 3



Separate Database per Tenant

Ogni tenant ha il proprio database separato

- Soluzione:

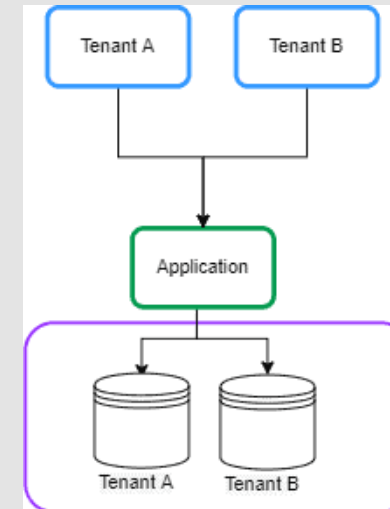
- Si mantiene a runtime una mappa tra i tenant e il rispettivo db
- Ogni tenant dispone di un utente configurato con privilegi limitati sul database su cui lavora

- Vantaggi:

- Livello più alto di sicurezza dei dati
- Isolamento delle performance e scalabilità
- Alto livello di personalizzazione del DB per il tenant

- Svantaggi:

- All'aumentare del numero di tenant aumenta anche la complessità del sistema
- Costi delle infrastrutture potenzialmente più alti
- Difficoltà nel replicare un cambiamento su più databases



Sprint 3



Implementazione (Separate Database per Tenant)

- Connessioni dinamiche personalizzate:

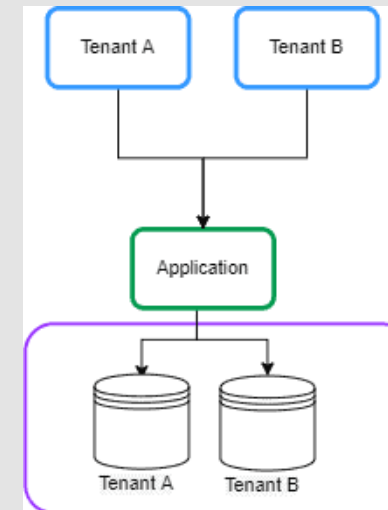
- La classe *DataSourceConfig* crea connessioni con credenziali specifiche per ogni tenant

- Gestione dinamica dei tenant:

- La classe *MultiTenantConnectionProvider* imposta il contesto SQL per identificare il corretto Database su cui operare

- Integrazione con Hibernate:

- Hibernate utilizza un *MultiTenantConnectionProvider* per gestire le connessioni multiple e un *CurrentTenantIdentifierResolver* per risolvere dinamicamente l'identificatore del tenant



Sprint 3



Tool di integrazione: analisi tramite Codacy, che al suo interno contiene anche ESLint

🕒 **Current** 1812 🕒 **Ignored** 0

🔍 Filter by Language ⌵ Severity ⌵ Category

All issues

<input type="checkbox"/>	Java	1085
<input type="checkbox"/>	Javascript	699
<input type="checkbox"/>	Python	26
<input type="checkbox"/>	YAML	2

Code patterns

Law Of Demeter 619

i18next-key-format 387

Avoid Trailing Comma 141

Clear Apply

🕒 **Current** 1812 🕒 **Ignored** 0

🔍 Filter by Language ⌵ Severity ⌵ Category

All issues

<input type="checkbox"/>	🟡 MEDIUM	1775
<input type="checkbox"/>	🔴 CRITICAL	31
<input type="checkbox"/>	🟢 MINOR	6

Code patterns

Law Of Demeter

i18next-key-format 387

Clear Apply

Sprint 3



Soft delete:

- Introdotta una classe astratta chiamata `SoftDeletableEntity`: per andare a filtrare le entità “soft deletable”, si è andati ad utilizzare i filtri di Hibernate, che in maniera automatica, permettono di aggiungere delle condizioni alla query sul DB.

```
@FilterDef(  
    name = "softDeleteFilter",  
    parameters = @ParamDef(name = "isDeleted", type = "boolean")  
)  
@Filter(name = "softDeleteFilter", condition = "is_deleted = :isDeleted")  
public abstract class SoftDeletableEntity {
```

- Questa classe deve essere estesa dalle entità che vogliono implementare la soft delete.

```
public class MedicalService extends SoftDeletableEntity {
```

```
public class Shift extends SoftDeletableEntity {
```


Sprint 3



Soft delete:

- Crea l'interfaccia `SoftDeleteJpaRepository` per estendere il comportamento del `JpaRepository`

```
@NoRepositoryBean 5 usages 3 implementations MatteoBasili
public interface SoftDeleteJpaRepository<T, ID> extends JpaRepository<T, ID> {
    @Override 1 implementation MatteoBasili
    void delete(T t);

    @Override 1 implementation MatteoBasili
    void deleteById(ID id);

    @Override 1 implementation MatteoBasili
    void deleteAll();

    @Override 1 implementation MatteoBasili
    void deleteAll(Iterable<? extends T> entities);

    void restoreById(ID id); no usages 1 implementation MatteoBasili
}
```

Sprint 3



Soft delete:

- Introdotto un nuovo filtro di Spring per l'attivazione della soft delete per ogni sessione
- Introdotta una annotazione @DisableSoftDelete a livello di singolo metodo. Per andare a disattivare il filtro in determinati metodi/casi, si è introdotta l'annotazione @DisableSoftDelete, il cui comportamento è catturato da una Aspect.

```
@Retention(RetentionPolicy.RUNTIME) 30
@Target(ElementType.METHOD)
public @interface DisableSoftDelete {
}
```

```
@Pointcut("@annotation(org.cswteams.ms3.config.annotations.DisableSoftDelete)")
public void disableSoftDeleteMethods() {}
```

Sprint 3



Stato dei turni:

- Per inserire uno stato dei turni si è lavorato con il componente Scheduler della libreria DevExtreme

Cardiologia Allocati: <ul style="list-style-type: none">FarnasiniVerdeSalvatiCantonePermessi	Cardiologia Allocati: <ul style="list-style-type: none">Cantone IIPermessiVerdeCantoneSalvati	Cardiologia Allocati: <ul style="list-style-type: none">CantoneVerdeSalvatiFarnasiniPermessiCantone II
--	---	---

P.s: per il futuro, se servirà nuovamente lavorare sul componente per gli appuntamenti, ricordarsi che è stato implementato in modo custom

Sprint 3



- **Esito:** lo sprint è stato quasi completato del tutto
 - Le due soluzioni di multi-tenancy e quella della soft delete sono state entrambe portate a raggiungimento
 - Il codice è stato analizzato sia frontend che backend, il secondo sia staticamente che dinamicamente
 - I test sono continuati ad essere sviluppati
 - Non si è riusciti ad introdurre il bottone per vedere in modo facile la lista di turni incompleti (Sprint successivi)

Sprint 3



- **Problemi riscontrati:**
 - Integrazione SonarCloud
 - L'integrazione di SonarCloud per la parte frontend ha avuto dei problemi per via della mancata esperienza, si è optato per Codacy, che integrava in automatico anche ESLint
 - Modifiche al componente Scheduler
 - Il componente scheduler utilizza due pattern, quello custom e quello in controlled mode, il primo che fa override del secondo, quindi da codice è facile confondersi.
 - Thread safety nella soluzione della Soft delete
 - Grave vulnerabilità di sicurezza col multi-tenancy
 - Le credenziali dei vari utenti database sono hard-coded

Sprint 3



Stato	Pesi	Descrizione
✓	3	<i>MultiTenant MultiSchema</i>
✓	3	<i>MultiTenant MultiDatabase</i>
✓	2	<i>Integrazione dei Tool di analisi: SonarCloud, ESLint, OWASPZap</i>
✓	5	<i>Introdurre la soft delete per i servizi</i>
✓	13	<i>Inserire lo stato dei turni sia nel backend che nel frontend con i colori per discriminarli</i>
✗	8	<i>Bottone per Planner per poter vedere in maniera facile la lista dei turni incompleti</i>
✓	3	<i>Schermata per la gestione dei turni per poterli visualizzare</i>
✓	8	<i>Permettere la cancellazione dei turni a livello grafico tramite softdelete</i>
✓	5	<i>Continuo ripristino dei test</i>

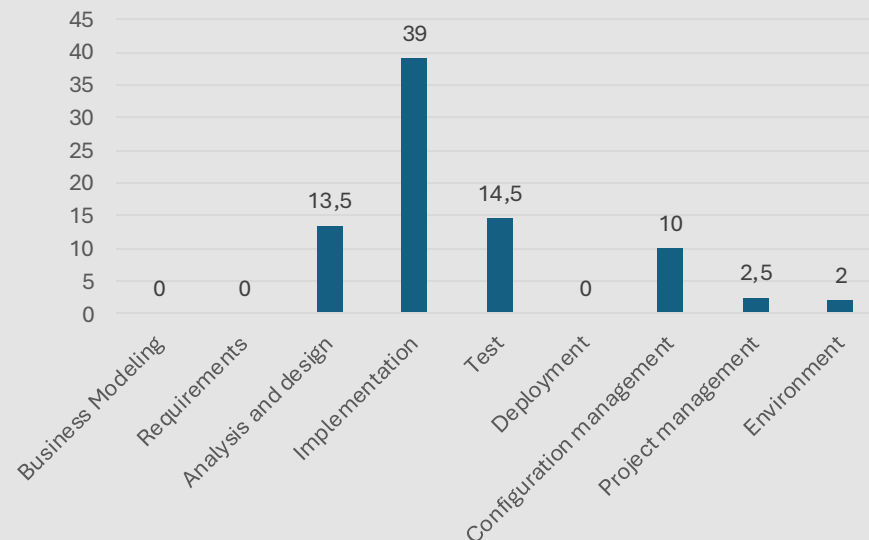
Tot done:	37
Tot:	45

INIZIO:	13/1/25
FINE:	27/1/25

Sprint 3



	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili			3	10	2		2	1,5		18,5	22,70%
Massimo Buniy				18	1		8	1		28	34,36%
Federico Cappellini				14	3					17	20,86%
Alessandro Finocchi			4,5	5	7			1,5		18	22,09%
Totale per disciplina	0	0	7,5	47	13	0	10	4	0	81,5	
Percentuale per disciplina	0,00%	0,00%	9,20%	57,67%	15,95%	0,00%	12,27%	4,91%	0,00%		



A photograph of a red running track with white lane lines. Several starting blocks are visible, arranged in a diagonal line across the frame. The blocks are silver with red and brown components. The text "Sprint 4" is overlaid in white on the track.

Sprint 4

27/01/2025 – 10/02/2025

Sprint 4



- **Durata:** 2 settimane
- **Periodo:** 27/01/2025 – 10/02/2025
- **Attività principali:**
 - Esternalizzare le credenziali di multi-tenancy nel file *application.properties*
 - Scegliere la soluzione di multi-tenancy di cui fare il merge
 - Documentare la scelta della soluzione di multi-tenancy
 - Introdurre OWASP Zap
 - Scrivere il documento di Risk Assessment

Sprint 4



Scelta soluzione Multi-tenancy

Si è scelto di adottare l'approccio **Single Database, Separate Schemas** per i seguenti motivi:

1. **Equilibrio tra isolamento e costi:** garantisce una buona segregazione dei dati senza richiedere la gestione di più database separati.
2. **Efficienza nella gestione:** la manutenzione e le operazioni di backup sono più semplici rispetto alla gestione di un database per ogni tenant.
3. **Scalabilità accettabile:** il modello può supportare un numero elevato di tenant con un'efficace gestione degli schemi.

Questa soluzione si è rivelata adatta alle nostre esigenze attuali, garantendo sicurezza, prestazioni e un costo infrastrutturale sostenibile.

Sprint 4



OWASP Zap report



DEPENDENCY-CHECK

Scan Information ([show all](#)):

- *dependency-check version*: 12.0.2
- *Report Generated On*: Sat, 15 Feb 2025 00:03:20 +0100
- *Dependencies Scanned*: 93 (71 unique)
- *Vulnerable Dependencies*: 26
- *Vulnerabilities Found*: 229
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
dom4j-2.1.1.jar	cpe:2.3:a:dom4j_project:dom4j:2.1.1:*:*:*:*:*	pkg:maven/org.dom4j/dom4j@2.1.1	CRITICAL	1	Highest	20
h2-1.4.197.jar	cpe:2.3:a:h2database:h2:1.4.197:*:*:*:*:*	pkg:maven/com.h2database/h2@1.4.197	CRITICAL	5	Highest	43
hibernate-core-5.4.2.Final.jar	cpe:2.3:a:hibernate:hibernate_orm:5.4.2:*:*:*:*:*	pkg:maven/org.hibernate/hibernate-core@5.4.2.Final	HIGH	2	Low	44
hibernate-validator-6.0.14.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.14:*:*:*:*:*	pkg:maven/org.hibernate.validator/hibernate-validator@6.0.14.Final	MEDIUM	3	Highest	32
jackson-annotations-2.9.0.jar	cpe:2.3:a:fasterxml:jackson-modules-java8:2.9.0:*:*:*:*:*	pkg:maven/com.fasterxml.jackson.core/jackson-annotations@2.9.0	MEDIUM	1	Low	37
jackson-databind-2.9.8.jar	cpe:2.3:a:fasterxml:jackson-databind:2.9.8:*:*:*:*:* cpe:2.3:a:fasterxml:jackson-modules-java8:2.9.8:*:*:*:*:*	pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.9.8	CRITICAL	54	Highest	41
log4j-api-2.11.2.jar	cpe:2.3:a:apache:log4j:2.11.2:*:*:*:*:*	pkg:maven/org.apache.logging.log4j/log4j-api@2.11.2	LOW	1	Highest	42
logback-classic-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-classic@1.2.3	HIGH	2	Highest	31
logback-core-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-core@1.2.3	HIGH	4	Highest	31

Sprint 4



Risk Assessment report

Storia delle revisioni.....	1
1 Introduzione.....	3
1.1 Scopo del report.....	3
1.2 Ambito della valutazione.....	3
1.3 Metodologia.....	3
2 Panoramica dell'applicazione.....	4
2.1 Descrizione dell'applicazione.....	4
2.2 Architettura del sistema.....	4
2.3 Funzionalità e caratteristiche chiave.....	4
3 Valutazione del rischio.....	5
3.1 Identificazione degli asset e dei dati sensibili.....	5
3.2 Identificazione delle minacce.....	7
3.4 Categorizzazione del rischio e prioritizzazione.....	10
4 Controlli di sicurezza.....	11
4.1 Misure di sicurezza attuali.....	11
4.2 Miglioramenti di sicurezza proposti.....	12
4.3 Allineamento con gli standard industriali.....	18
5 Strategie di mitigazione del rischio.....	19
5.1 Piano di rimedio proposto.....	20
5.2 Accettazione del rischio e rischio residuo.....	22
6 Risposta e recovery ad incidenti.....	23
6.1 Monitoraggio e rilevamento degli incidenti.....	23
6.2 Piano di risposta.....	24
Appendice A - Acronimi e glossario.....	26

Sprint 4



- **Esito:** lo sprint è stato soggetto a rallentamenti, è stato etichettato come completato e si è passati allo sprint successivo.
 - Chiuso il discorso del multi-tenancy, esternalizzando le credenziali, scritta la documentazione della motivazione della scelta e fatto il merge della soluzione scelta.
 - Introdotta il tool OWASP Zap, si è deciso di utilizzarlo come tool esterno, senza automatizzarlo.
 - Incominciato a scrivere il documento di Risk Assessment.

Sprint 4



- **Problemi riscontrati:**
 - Automatizzazione OWASP Zap
 - Nel cercare di integrare OWASP Zap nel ciclo di build del progetto si sono trovate parecchie difficoltà, così come nella sua integrazione con GitHub
 - Il merge del multi-tenancy ha portato i test che si erano ripristinati a dover nuovamente cambiare

Sprint 4



Stato	Pesi	Descrizione
✓	1	Esternalizzazione delle credenziali sull'application properties
✓	3	Scegliere quale soluzione di multitenancy mergare nel main
✓	2	Documentazione sulla scelta della soluzione di multitenancy
✓	2	Introdurre OWASP Zap
✗	8	Bottone per Planner per poter vedere in maniera facile la lista dei turni incompleti
✗		Ripristino dei test
✗	21	Preservare il principio di anonimizzazione
		Sicurezza dei dati (data protection): cifratura del sistema operativo, è un meccanismo che rende il disco cifrato se acceduto su altre macchine e in chiaro sulla nostra per evitare di dover cifrare/decifrare ad ogni query. Cercare soluzioni per docker
		Risk Assesment -> lista di interventi da applicare al prossimo Sprint, con tool, rischi e criticità. iniziare la lavorazione degli aspetti più critici

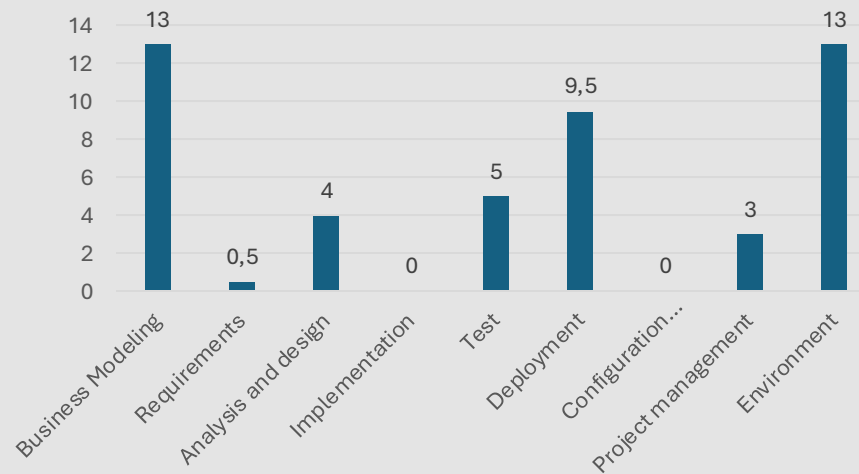
Tot done:	8
Tot:	37

INIZIO:	27/1/25
FINE:	10/2/25

Sprint 4



	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili		1,5	2		2		5	1		11,5	24,47%
Massimo Buniy	3				2		6	1		12	25,53%
Federico Cappellini			4				5,5	1		10,5	22,34%
Alessandro Finocchi	7,5		1			3,5		1		13	27,66%
Totale per disciplina	10,5	1,5	7	0	4	3,5	16,5	4	0	47	
Percentuale per disciplina	22,34%	3,19%	14,89%	0,00%	8,51%	7,45%	35,11%	8,51%	0,00%		





Sprint 5

10/02/2025 – 24/02/2025

Sprint 5




- **Durata:** 2 settimane
- **Periodo:** 10/02/2025 – 24/02/2025
- **Attività principali:**
 - Implementata una funzionalità per il Planner
 - Scrittura del documento sulle scelte per l'analisi statica e dinamica del codice
 - Completamento del Risk Assessment Report





Sprint 5





Visualizzazione turni come lista


 Menu

 Turni globali

 I miei turni

 Genera Pianificazione

 Utenti

 Scambio Turni

MEDICAL STAFF SHIFT SCHEDULER - B

AGGIUNGI TURNO CONCRETO

SCARICA QUESTI TURNI COME CSV

CALENDARIO

LISTA IMPEGNI

FILTER

MAR 25 FEB	2:00 – 8:00	Cardiologia
MAR 25 FEB	9:00 – 15:00	Cardiologia
MAR 25 FEB	16:00 – 22:00	Cardiologia
MER 26 FEB	2:00 – 8:00	Cardiologia
MER 26 FEB	16:00 – 22:00	Cardiologia
GIO 27 FEB	2:00 – 8:00	Cardiologia
GIO 27 FEB	16:00 – 22:00	Cardiologia

Sprint 5



- **Esito:** Lo sprint è andato a buon fine
 - Il risk assessment è stato completato
 - Il documento riguardante le scelte fatte per l'analisi statica e dinamica è stato scritto
 - Si è aggiunta la possibilità di vedere tutti i turni come una lista (google-calendar-like)
 - Il pianificatore può vedere la lista dei soli turni incompleti generati

Sprint 5



Stato	Pesi	Descrizione
✓		OWASP Zap: documento
✓		Bottone per Planner per poter vedere in maniera facile la lista dei turni incompleti
✓		Risk Assesment -> completare il documento

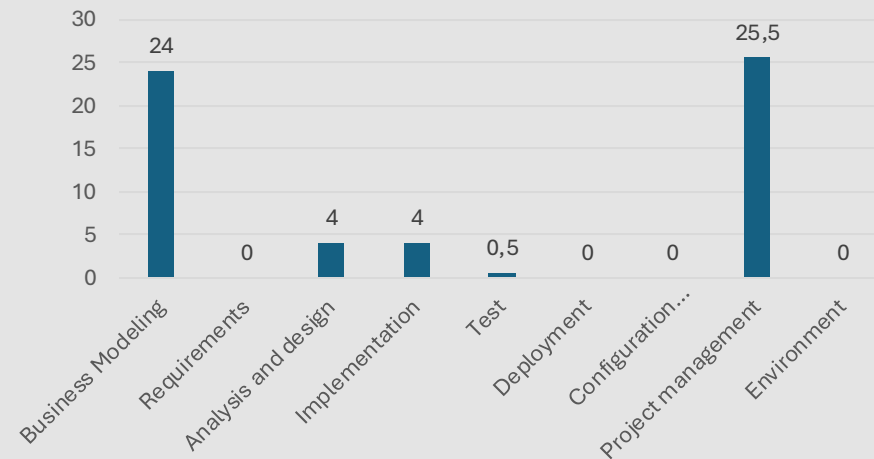
INIZIO: 10/2/25

FINE: 24/2/25

Sprint 5



	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili	5,5				0,5			8,5		14,5	26,13%
Massimo Buniy	3		4	3				6,5		16,5	29,73%
Federico Cappellini	4									4	7,21%
Alessandro Finocchi	11,5							9		20,5	36,94%
Totale per disciplina	24	0	4	3	0,5	0	0	24	0	55,5	
Percentuale per disciplina	43,24%	0,00%	7,21%	5,41%	0,90%	0,00%	0,00%	43,24%	0,00%		



Analisi della produzione

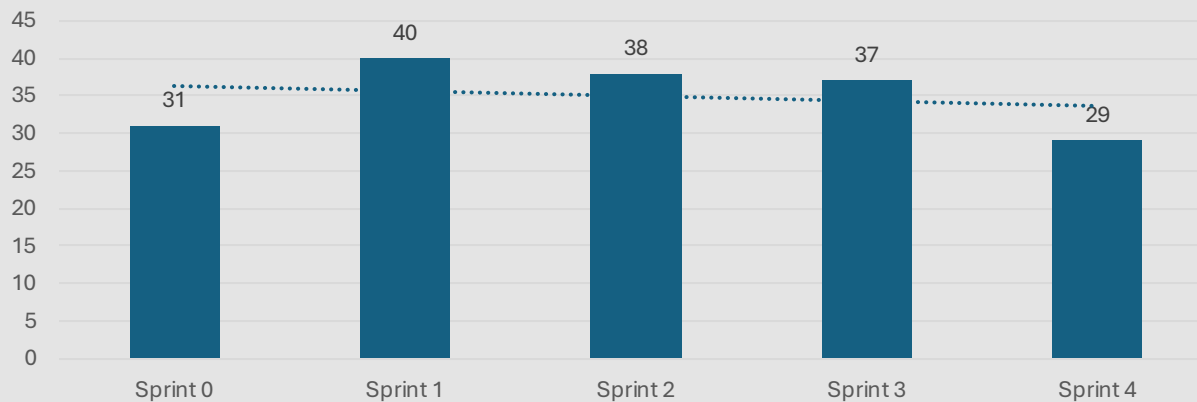


Analisi della produzione



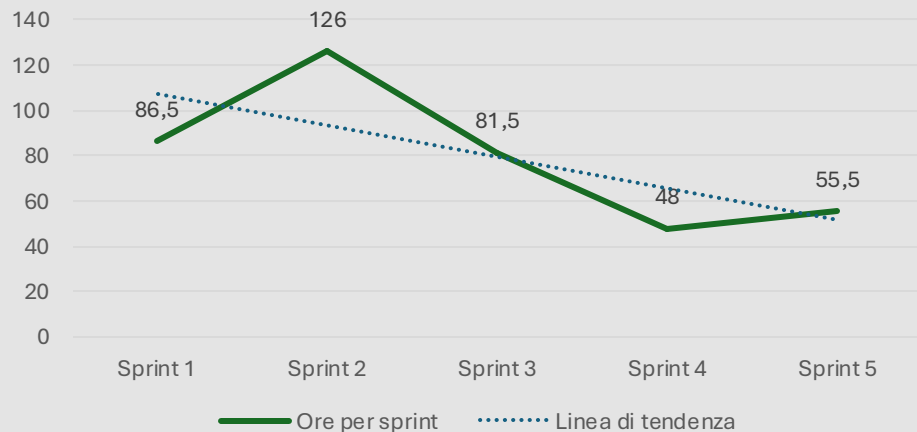
Studio dell'andamento dei function points attraverso ogni sprint

Function points per sprint

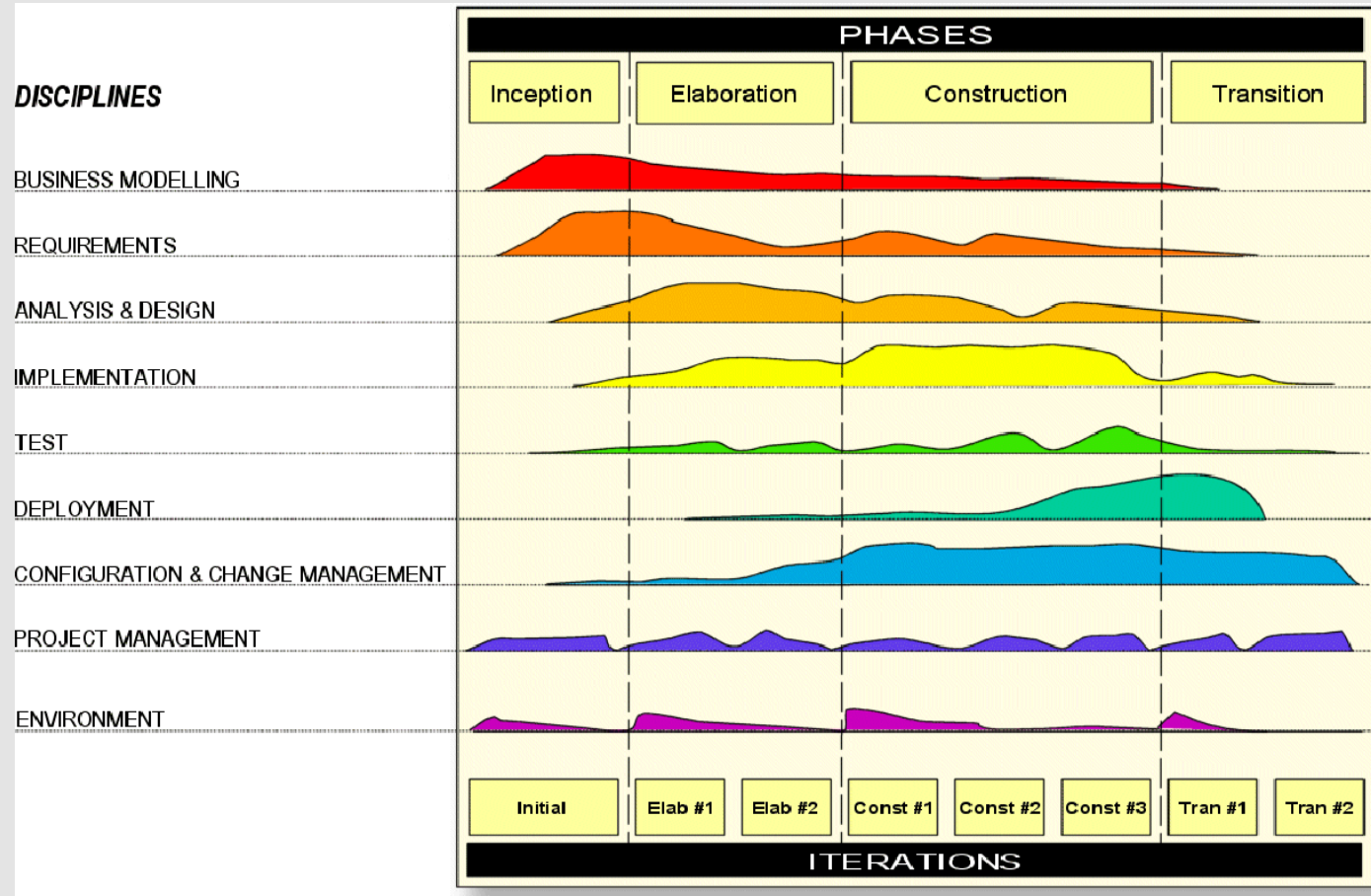


Media finale tra gli sprint:
35 FP per sprint

Grafico ore complessive spese per sprint



Modello RUP

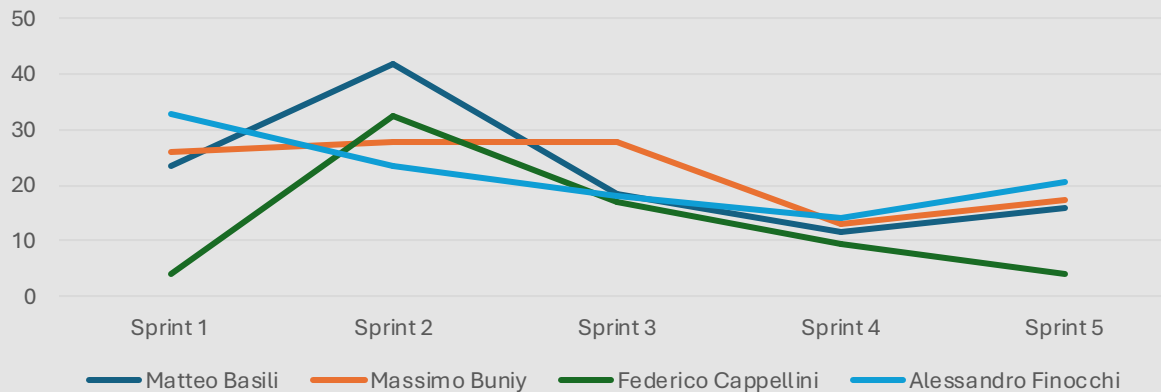


Analisi della produzione

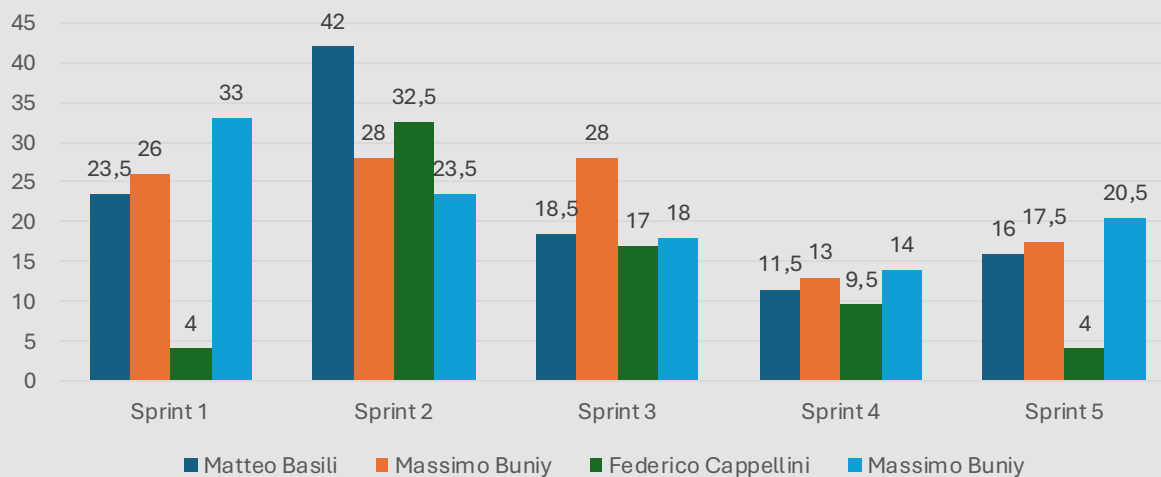


Grafico dell'andamento delle ore tra i vari sprint

Andamento ore del singolo per Sprint



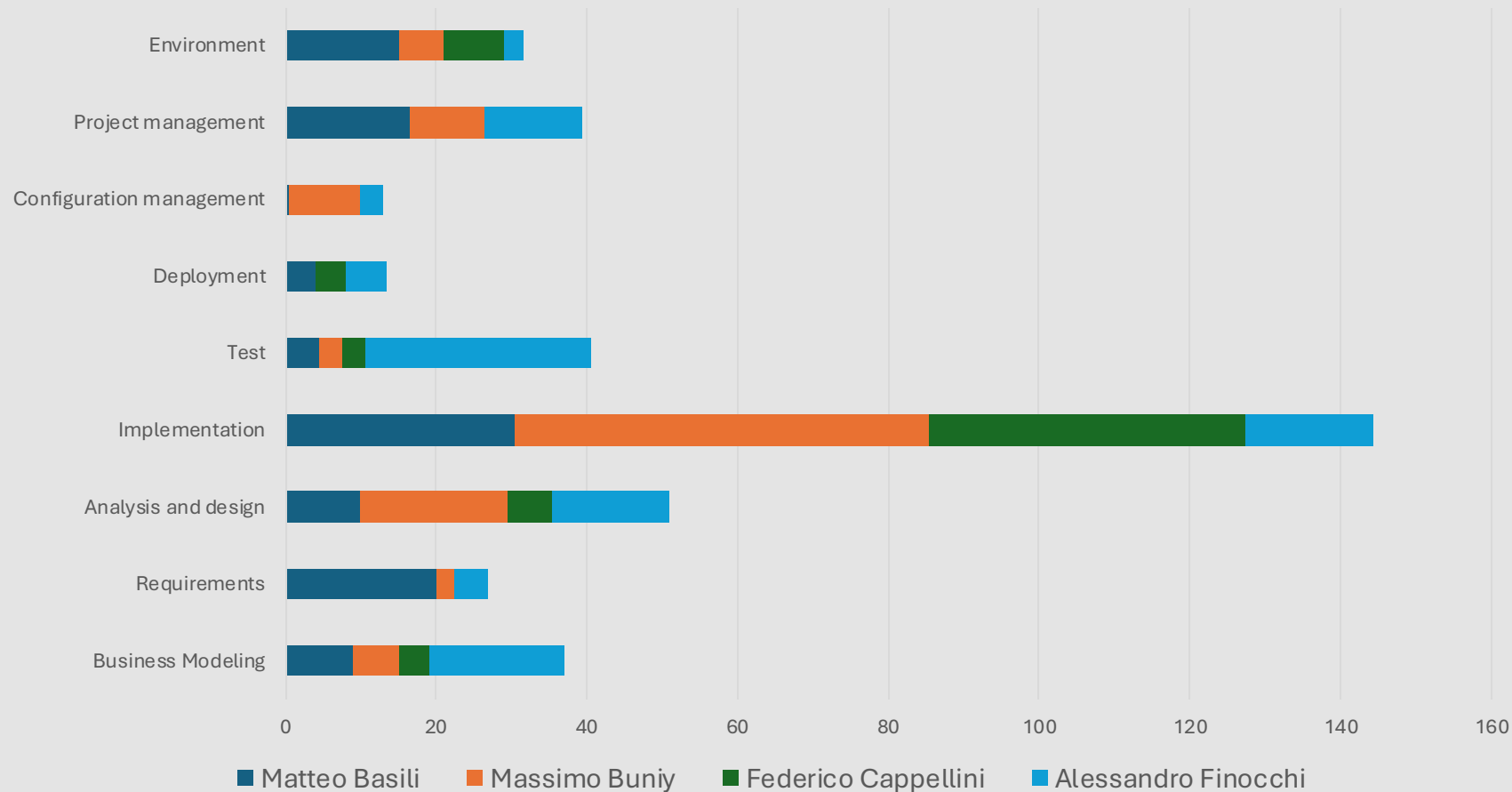
Andamento ore del singolo per Sprint



Analisi della produzione



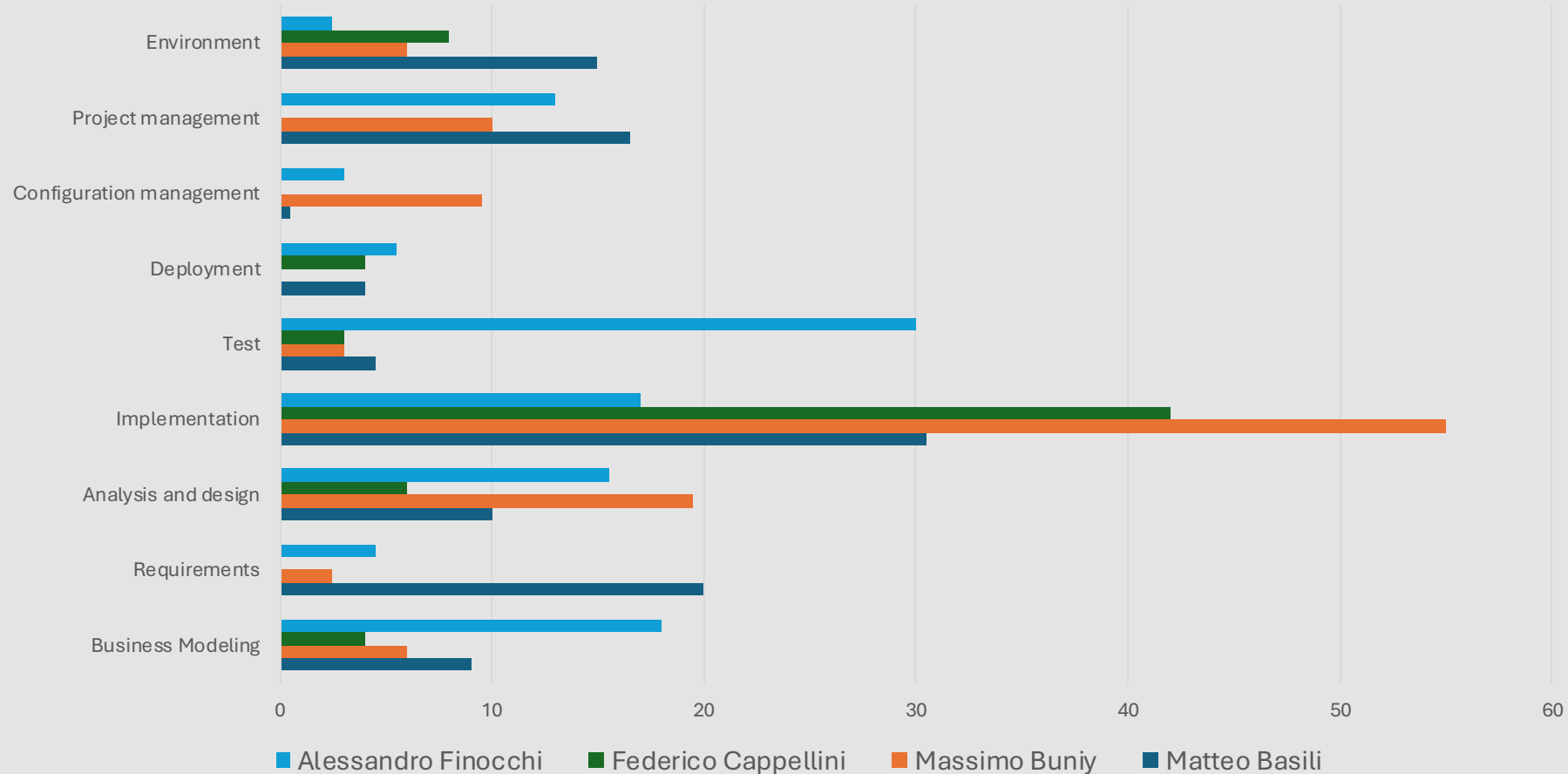
Effort sulla singola Area di Processo



Analisi della produzione



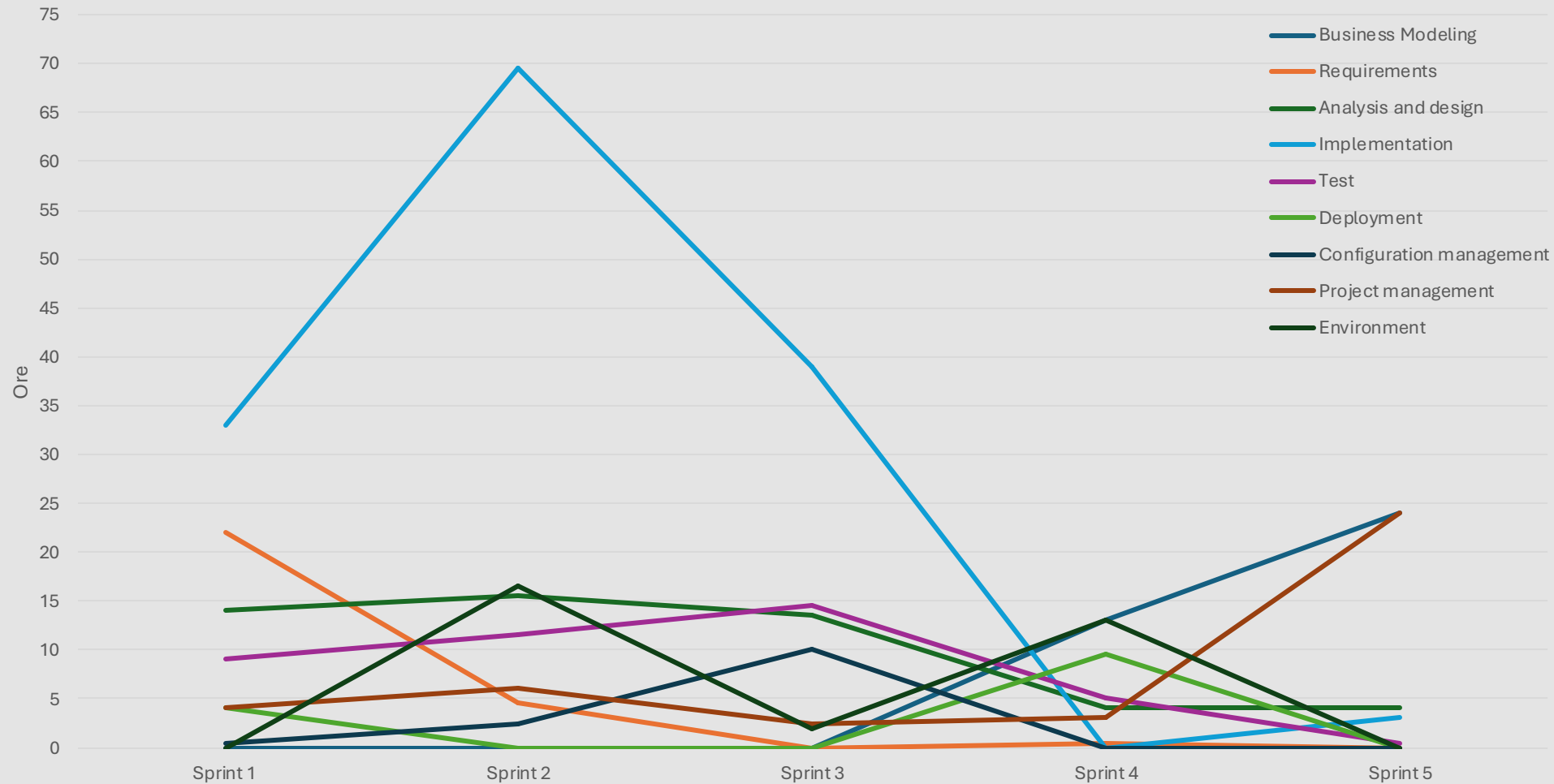
Effort del singolo a confronto sulla singola area di processo



Analisi della produzione



Andamento delle ore nelle Aree di Processo durante i vari Sprint



Analisi della produzione

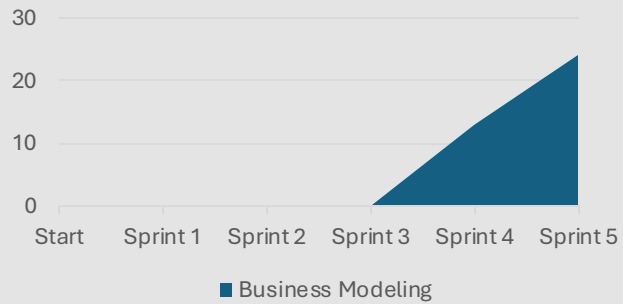


	Business Modeling	Requirements	Analysis and design	Implementation	Test	Deployment	Configuration management	Project management	Environment	Totale	Percentuale
Matteo Basili	9	20	10	30.5	4.5	4	0.5	18	15	111.5	27.88%
Massimo Buniy	6	2.5	19.5	56	3	0	9.5	10	6	112.5	28.13%
Federico Cappellini	4	0	6	42	3	4	0	0	8	67	16.75%
Alessandro Finocchi	18	4.5	15.5	17	30	5.5	3	13	2.5	109	27.25%
Totale per disciplina	37	27	51	145.5	40.5	13.5	13	41	31.5	400	
Percentuale per disciplina	9.25%	6.75%	12.75%	36.38%	10.13%	3.38%	3.25%	10.25%	7.88%		

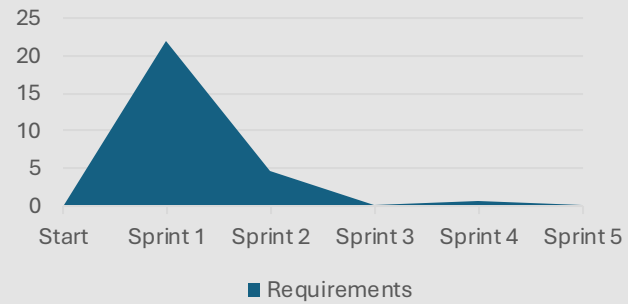
Analisi della produzione



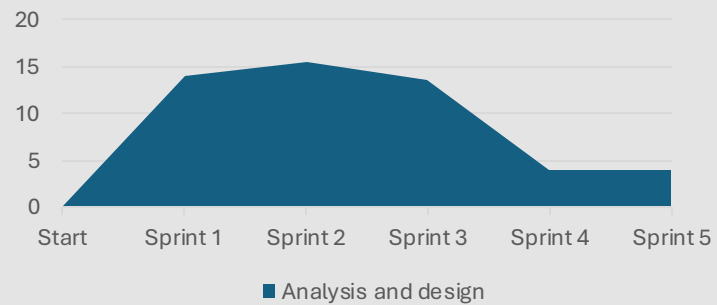
Business Modeling



Requirements



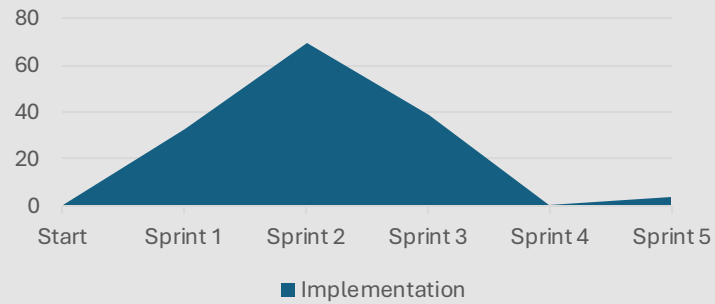
Analysis and design



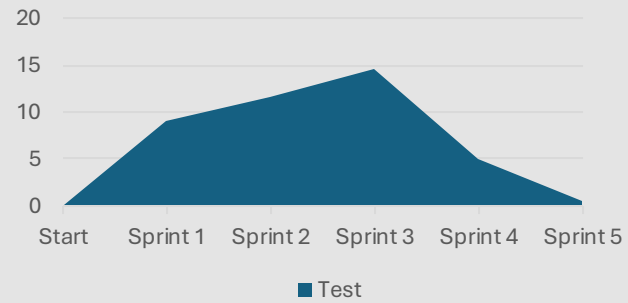
Analisi della produzione



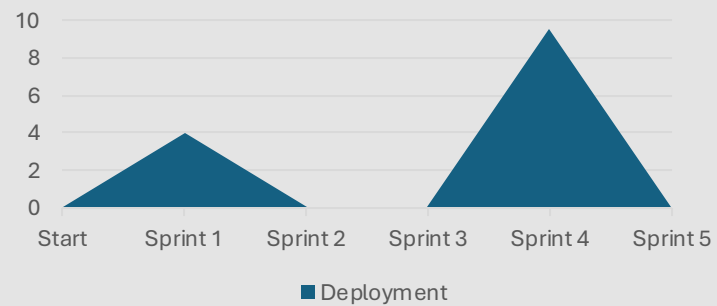
Implementation



Test



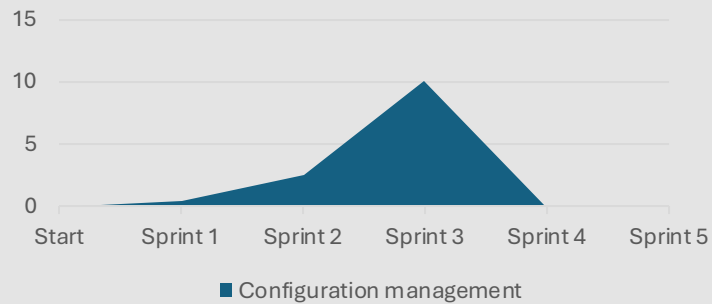
Deployment



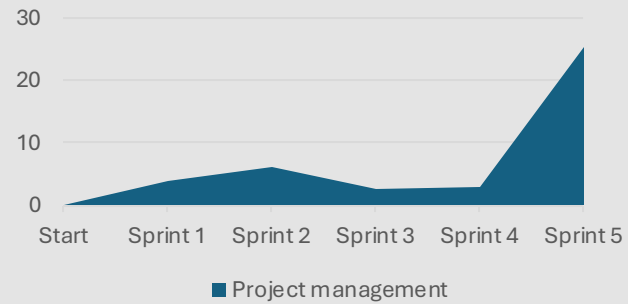
Analisi della produzione



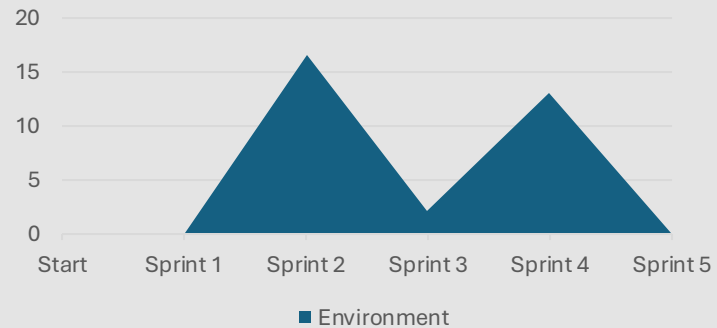
Configuration management



Project management



Environment



Analisi della produzione



- Gli sprint sono stati incentrati al raggiungimento dei goal
- L'analisi comparativa con RUP non trova una congiunzione dal momento gli incrementi dell'applicazione erano incentrati sulle nuove funzionalità non progettate originariamente, per cui il tempo è stato speso nell'adattarsi a ciò
- Picchi iniziali di implementation indicano proprio questo fenomeno: ci si è concentrati su poche funzionalità ma impattanti nell'utilizzo del tempo
- Verso la fine l'effort preponderante è stato in direzione della documentazione, a sottolineare un comportamento non ciclico e ripetitivo dell'incremento

A close-up photograph showing a hand in a dark suit sleeve holding a lit torch. The torch has a dark, textured head and a bright orange and yellow flame. Below the torch, another hand in a similar suit sleeve is held open, palm up. A thin white horizontal line is positioned across the middle of the image, passing behind the text. The background is a soft-focus blue and white sky with clouds.

Ai posteri

Ai posteri



- Lo scopo di questa sezione è evidenziare quali siano le tematiche e le questioni più importanti da lasciare a coloro che lavoreranno al progetto dopo di noi



Documentazione

- Estesa su [MS3-docs](#)
 - Documento di analisi sulle soluzioni di **Multi-Tenancy**, inclusa la scelta dell'architettura più adatta per la segregazione delle informazioni tra i tenant
 - Documento sulle **API Authorizations**
 - **Risk Assessment Report**
 - Documento sui risultati dell'analisi **statica** e **dinamica** del codice



Task più rilevanti orientati alle funzionalità

- Aggiornare diagramma E-R ([#17](#))
- Ripristino test considerando le modifiche del refactoring e l'introduzione del Multi-tenancy ([#610](#))
- Risolvere le incoerenze del glossario ([#619](#))
- Gestione dei warning a runtime sulla console del browser ([#620](#))
- Logout non implementato ([#621](#))
- Differenziare l'assegnazione degli uffa points di un dottore di guardia da quelli di uno reperibile (devono avere diverso peso) ([#622](#))

Ai posteri



Nuovi task orientati alla sicurezza

- Cifratura del sistema operativo ([#603](#), [#618](#))
- Anonimizzazione dei dati ([#623](#))
- Implementazione 2FA ([#624](#))
- Validazione dell'input ([#625](#))
- Aggiunta sistema di logging e audit trail ([#626](#))
- Connessione sicura ([#627](#))

CONCLUSIONI

**GRAZIE PER LA VOSTRA
ATTENZIONE!**

Link repository:



<https://github.com/CSW-Teams>