

Università degli studi di Roma Tor Vergata
Ingegneria dei Sistemi Software e dei Servizi in Rete
A.A. 2015/2016

GQM+S Tool - Phase 3 & 4 subphase 2

Prof. Giovanni Cantone

Team members:

Gabriele Belli

Daniele Capri

Giuseppe Chiapparò

Fabio Alberto Coira

Luca Della Gatta

Federico Di Domenicantonio

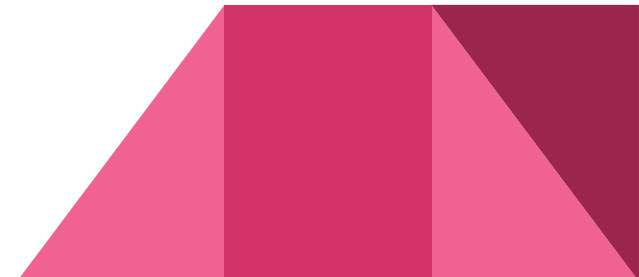
The GQM+Strategies Approach



GQM+Strategies

GQM+S è un approccio per allineare gli obiettivi e le strategie di una organizzazione attraverso misure.

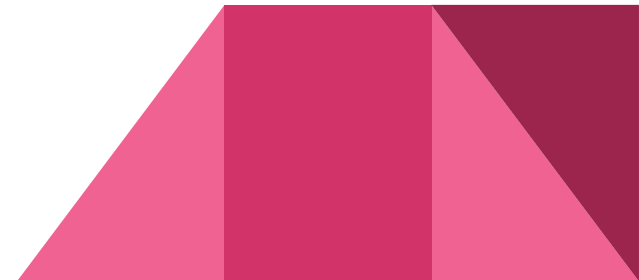
I “**Goals**” sono gli stati futuri che l’azienda vuole raggiungere mentre le “**Strategies**” sono l’inseme di tutte quelle azioni da compiere per raggiungere gli obiettivi prefissati.



Durante il processo vengono raccolti dati attraverso apposite **misurazioni** al fine di valutare sistematicamente il **raggiungimento degli obiettivi**.

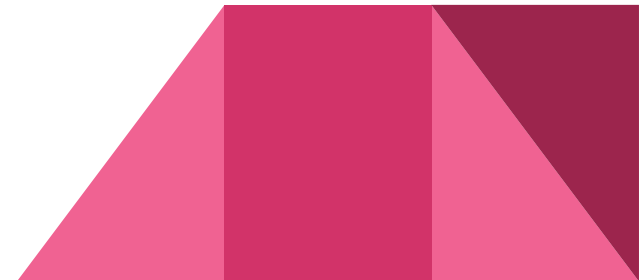
Il principale risultato dell'utilizzo di questo approccio deriva dal fatto che ogni decisione dell'organizzazione sarà corretta e migliorata in funzione dei **dati misurati**.

Per effettuare le misurazioni devono essere definiti modelli di misura: in questo contesto ad ogni obiettivo organizzativo (**organization goal**) deve essere associato un obiettivo di misura (**measurement goal**).

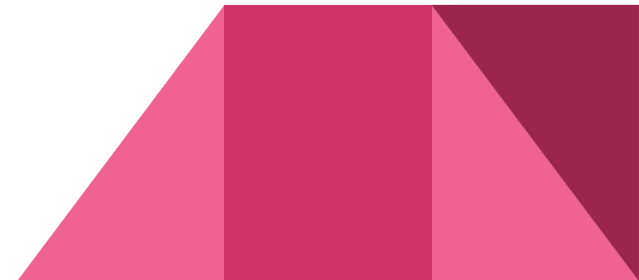


Per ogni measurement goal è definito un modello interpretativo (**interpretation model**) che supporta la valutazione delle misure effettuate.

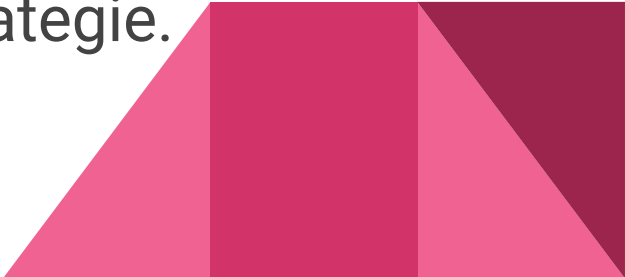
Fattori di contesto ed ipotesi sono necessari in quanto forniscono un set di obiettivi e strategie scelti in maniera razionale rispetto ad una specifica organizzazione ed al suo ambiente organizzativo.



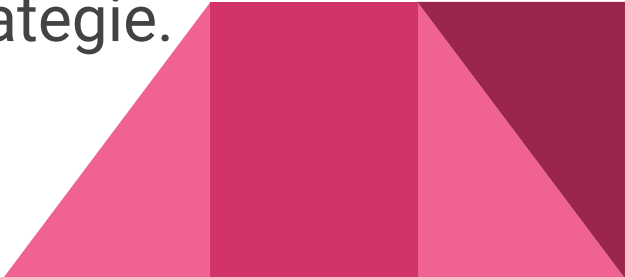
Tipicamente il set di obiettivi e strategie definiti inizialmente introducono obiettivi e strategie ad un livello più basso: l'intero modello gerarchico è chiamato griglia (**grid**).



Il processo applicativo dell'approccio GQM+S consiste in sei fasi, le quali possono essere ripetute ciclicamente con una certa frequenza:

- **Sviluppo:** viene sviluppato il modello e la GQM grid.
 - **Implementazione:** prevede la messa in atto delle strategie sviluppate nella fase precedente. Vengono inoltre eseguite le misurazioni necessarie a verificare il raggiungimento degli obiettivi proposti.
 - **Apprendimento:** vengono analizzati i risultati ottenuti dall'applicazione delle strategie al fine di migliorare il processo e generare nuovi obiettivi e strategie.
- 

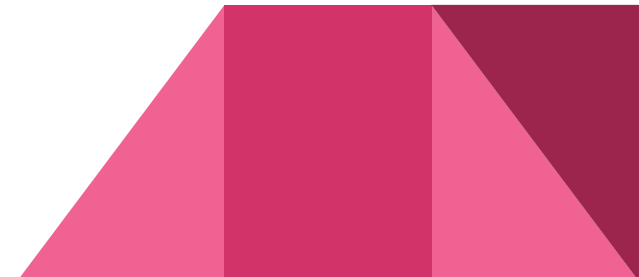
Il processo applicativo dell'approccio GQM+S consiste in sei fasi, le quali possono essere ripetute ciclicamente con una certa frequenza:

- **Sviluppo:** viene sviluppato il modello e la GQM grid.
 - **Implementazione:** prevede la messa in atto delle strategie sviluppate nella fase precedente. Vengono inoltre eseguite le misurazioni necessarie a verificare il raggiungimento degli obiettivi proposti.
 - **Apprendimento:** vengono analizzati i risultati ottenuti dall'applicazione delle strategie al fine di migliorare il processo e generare nuovi obiettivi e strategie.
- 

Fase 3: Attuazione della griglia di pianificazione

Obbiettivo della fase: Specificare i piani per rendere operativa la griglia di GQM+Strategies nell'organizzazione.

Questo significa: la preparazione per l'attuazione e l'implementazione di strategie organizzative definite nella griglia e per la raccolta di analisi e interpretazione dei dati secondo le metriche e modelli interpretativi.



Inputs:

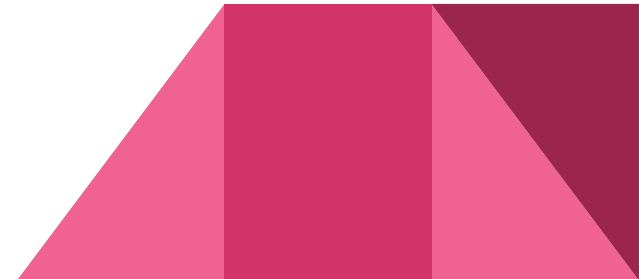
- Griglia di GQM + Strategies .
- Documentazione dei processi di lavoro in corso.

Attività:

- Sviluppo piani strategici.
- Sviluppo piani di misura.
- Formazione del personale.

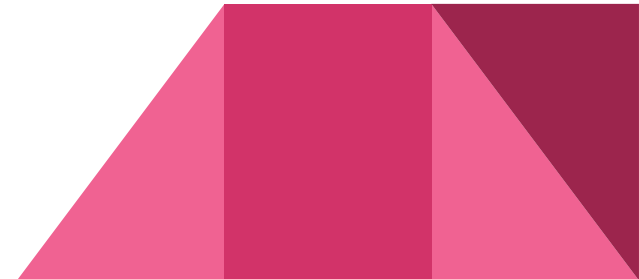
Risultati:

- Piani strategici.
- Piani di misura.



Nel nostro contesto di subphase 2 la fase 3 si articola:

- ❑ **Input** : Strategie terminali dal bus.
- ❑ **Attività:**
 - ❑ Sviluppo dello Strategic Plan.
 - ❑ Pianificazione della Validazione.
- ❑ **Risultati:**
 - ❑ Piano Strategico.
 - ❑ Piano di Validazione.

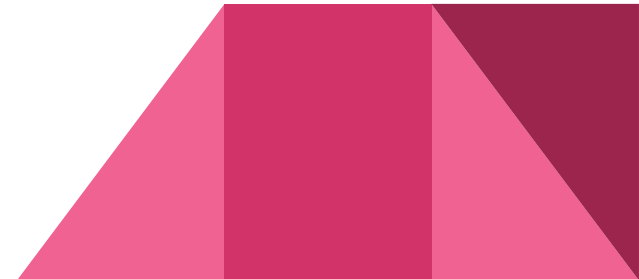


Fase 4: Eseguire il modello

Obbiettivo della fase: Applicare le strategie e analizzare le loro prestazione e fornire un feedback.

Inputs:

- Griglia di GQM + Strategies.
- Piani Strategici.
- Piani di Misura.

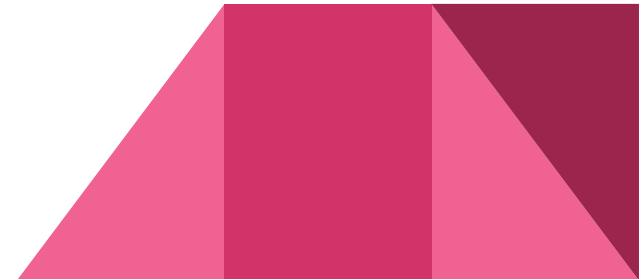


Attività:

- Eseguire le strategie.
- Collezionare e analizzare i dati.
- Fornire un feedback.

Risultati:

- Feedback.
- Il risultato delle strategie.
- Dati validati.



Nel nostro contesto di subphase 2 la fase 4 si articola:

- ❑ **Input** : Piano di validazione.
- ❑ **Attività:**
 - ❑ Esecuzione della Validazione.
- ❑ **Risultati:**
 - ❑ Dati validati.



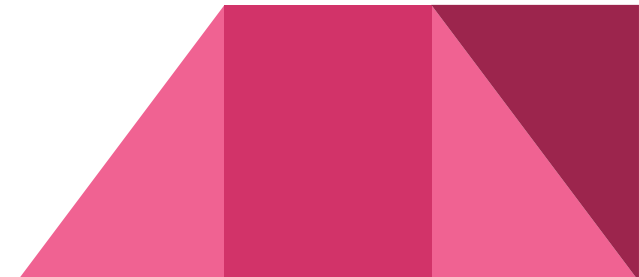


Approccio adottato: Scrum

Agile

È stato deciso di utilizzare Scrum:

- Framework di best practices
- Iterativo ed incrementale
- Adatto a cambi di requisiti



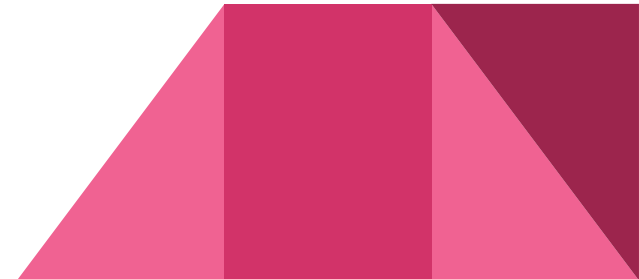
Scrum: trasparenza

Linguaggio comune del processo:

- Termini di Scrum
- Glossario

Definition of done:

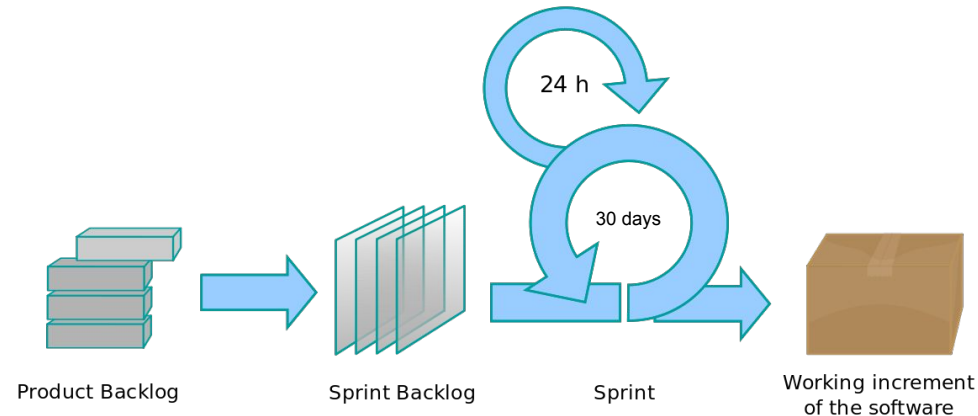
- Feature
 - Test unitari
 - Test integrazione



Scrum

Caratteristiche:

- Team di sviluppo
 - Gabriele Belli
 - Daniele Capri
 - Giuseppe Chiappararo
 - Fabio Alberto Coira
 - Luca Della Gatta
 - Federico Di Domenicantonio
- Scrum master
 - Gianfranco Pesce
- Product owner
 - Giovanni Cantone



Scrum

Sprint planning meeting

- 7 giorni

Backlog Grooming

Scrum of scrums

- 7 giorni



Scrum

Daily scrum

- Incontri quotidiani
- Interazione telematica
 - Slack
 - Telegram

Sprint review

- Coincide con lo sprint planning meeting

Sprint retrospective



Scrum

Back-end

- **164** commits during **63** days
- Average **2.6** commits per day
- Contributed by **8** authors

Front-end

- **196** commits during **28** days
- Average **7.0** commits per day
- Contributed by **14** authors

[Eseguiti su gitlab](#)

[Eseguiti su gitlab](#)



GitLab












Scrum

Storie effettuate:

- 23 Storie

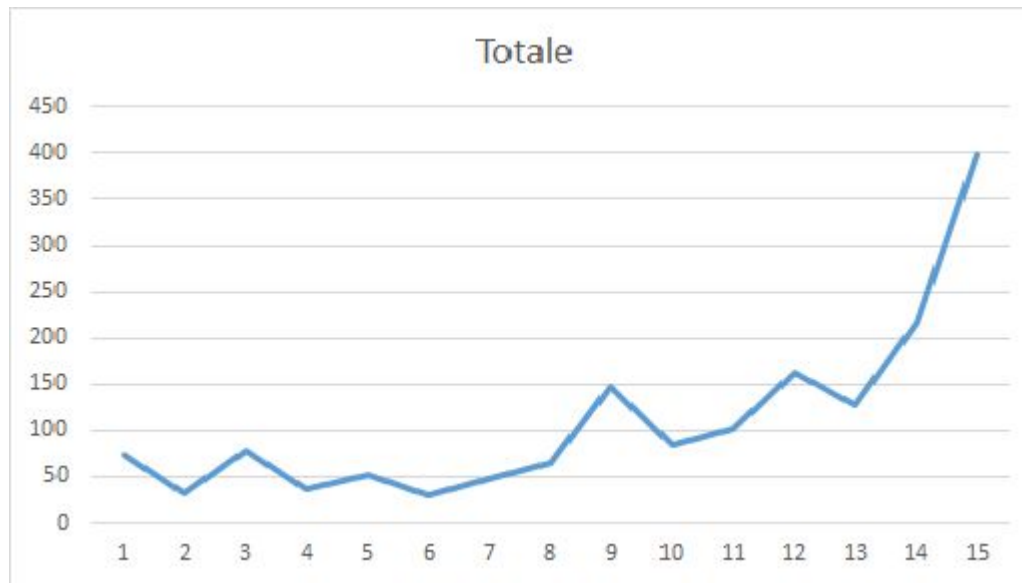
Task effettuati:

- 58 Task

Product Backlog	Sprint Backlog	In Progress	Peer Review	In Test	Done	Blocked
						
						
						
						
						

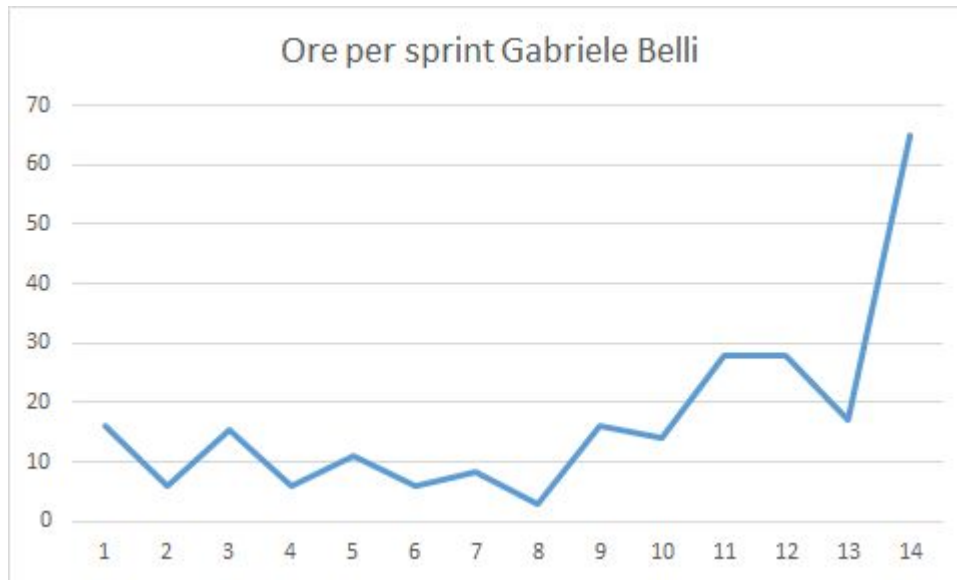
Scrum: ore totali

Totale ore: 1715,5



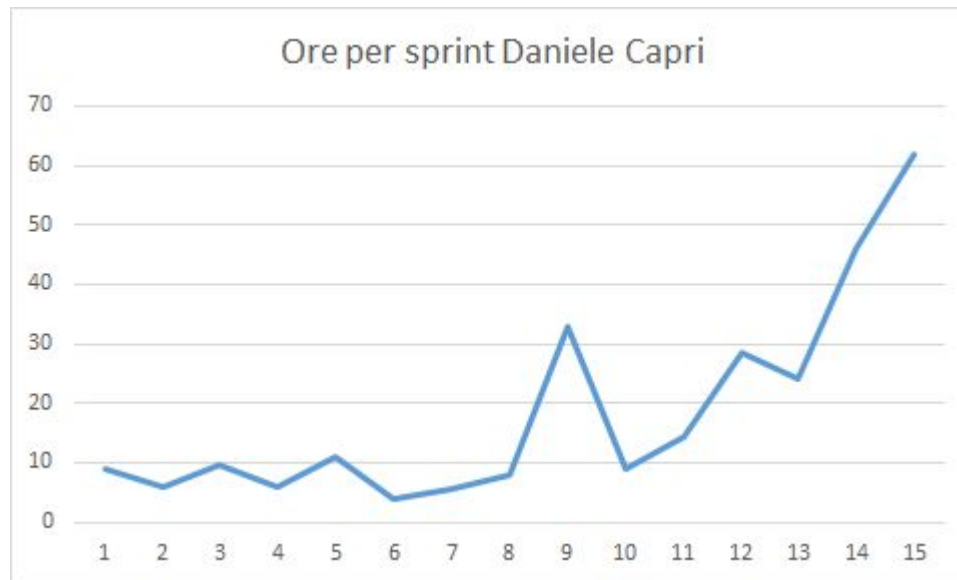
Scrum: ore totali Gabriele Belli

Totale ore: 250



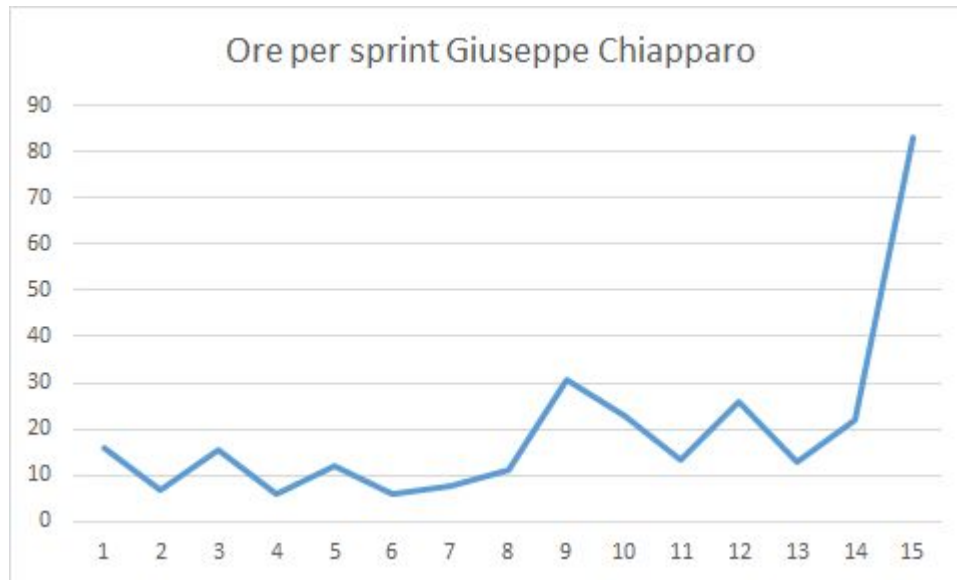
Scrum: ore totali Daniele Capri

Totale ore: 275,5



Scrum: ore totali Giuseppe Chiapparo

Totale ore: 292



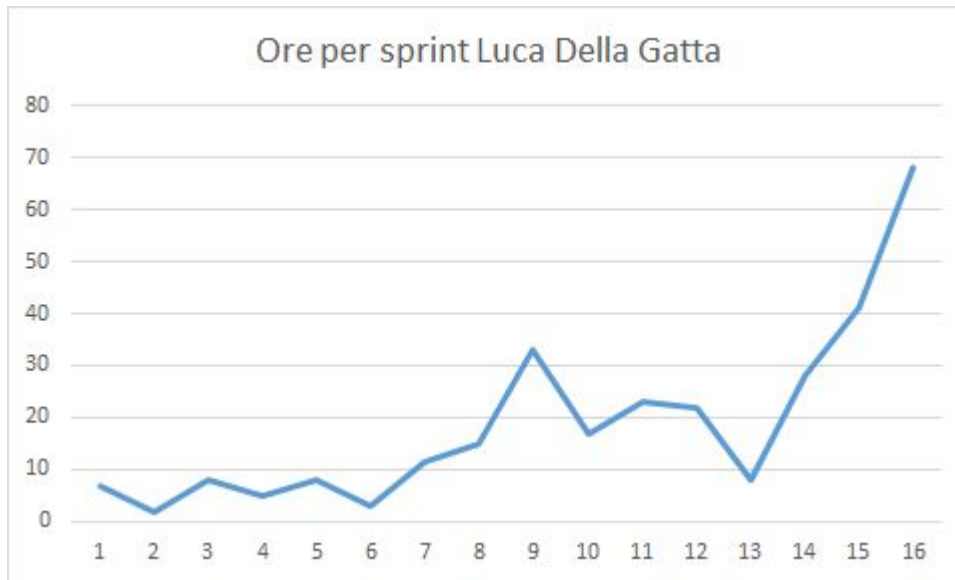
Scrum: ore totali Fabio Alberto Coira

Totale ore: 322



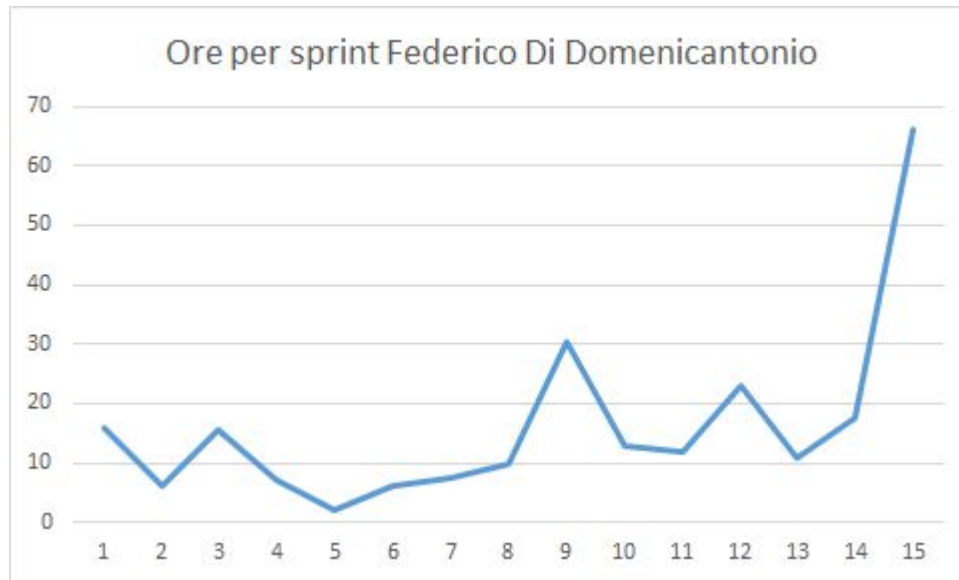
Scrum: ore totali Luca Della Gatta

Totale ore: 291



Scrum: ore totali Federico Di Domenicantonio

Totale ore: 285,5

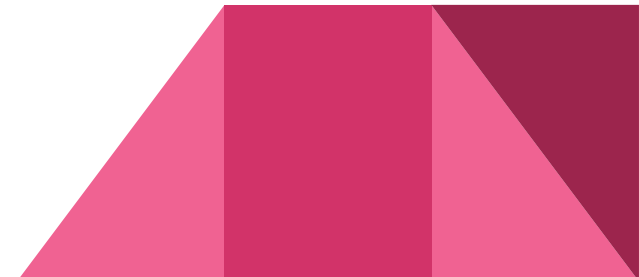




Integrazione con gli altri gruppi

Fase 3.2 4.2

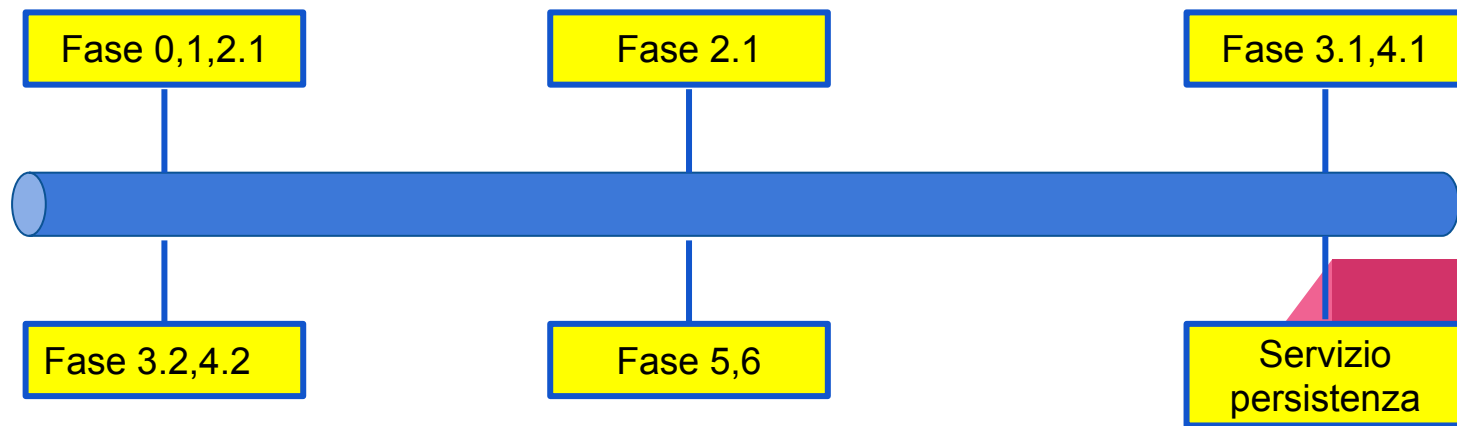
- Input: Strategie terminali
 - Id
 - Nome
 - Breve Descrizione
 - Unità organizzativa
 - Rivisitato
 - Versione
- Output: Dati validati
 - Id del dato
 - Set di dati validati
 - Id dell'istanza di business di workflow
 - Riferimento alla strategia
 - Riferimento alla metrica



ESB - Enterprise service bus

Infrastruttura software che fornisce servizi supporto

- Instradamento intelligente
- Coordinamento
- Messaggistica



Integrazione con gli altri gruppi

Fase 1:

- Input: **Strategie foglia**

Cosa ci aspettiamo:

```
{
  "description": "Description sample strategy",
  "id": 1, "state": {}, "title": "Sample Strategy",
  "projectId": 2,
  "subStrategies": []
}
```

```
"tags": [{ "value" : "notYetLinked" , "key" : "OrgUnitId" },
           { "value" : "notYetLinked" , "key" : "OrgUnitName" }]
```

Cosa ci arriva:

```
1 ▾ {"tag": "level3Direct",
2 ▾  "content": [
3   "[{"value\\":\\"notYetLinked\\",\\"key\\":\\"OrgUnitName\\"}], {"busVersion\\":\\"4\\",\\"instance\\":\\"Sample Strategy\\",\\"err\\":\\"0\\",\\"payload\\":\\"{"description\\":\\"Description sample strategy\\",\\"id\\":1,\\"state\\":{}\\",\\"title\\":\\"Sample Strategy\\",\\"projectId\\":2,\\"subStrategies\\":[]}\\",\\"typeObj\\":\\"base64-TerminalStrategy\\",\\"tags\\":[{\\"value\\":\\"notYetLinked\\",\\"key\\":\\"OrgUnitId\\"},{\\"value\\":\\"notYetLinked\\",\\"key\\":\\"OrgUnitName\\"}]}"
4   ],
5   "resolvedAddress": "",
6   "originAddress": "",
7   "id": null
8  }
```

Integrazione con gli altri gruppi

Fase 5:

- Output: *Set di dati validati*

Cosa si aspettano:

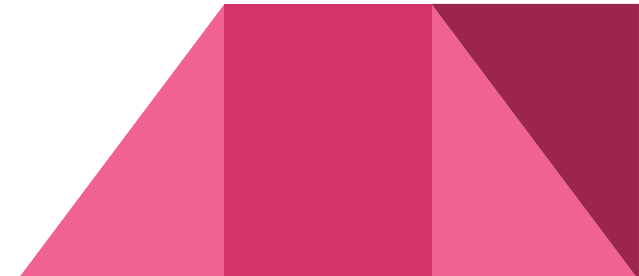
```
{
  "dataId": "1",
  "data": ["1448", "48", "1315", "157", "1442", "360", "846", "871", "565", "1109", "695", "332", "1273", "1338", "775", "1093", "867", "308", "1082", "1239", "1020", "1163", "431", "1379", "772", "1030", "1051", "304"],
  "businessWorkflowInstanceId": "1",
  "strategyRef": "18",
  "metricRef": "1"
}
```

Cosa arriva dal bus:

```
1 {
2   "tag": "level3Direct",
3   "content": [
4     [{"busVersion": "1", "instance": "3749", "err": "0", "payload": {"data": {}, "dataId": "3749", "businessWorkflowInstanceId": "56", "metricRef": "7564", "strategyRef": "64795"}, "typeObj": "validateData", "tags": []}, {"busVersion": "1", "instance": "3749", "err": "0", "payload": {"data": {}, "dataId": "3749", "businessWorkflowInstanceId": "56", "metricRef": "7564", "strategyRef": "64795"}, "typeObj": "validateData", "tags": []}, {"busVersion": "2", "instance": "3749", "err": "0", "payload": {"data": {}, "dataId": "3749", "businessWorkflowInstanceId": "56", "metricRef": "7564", "strategyRef": "64795"}, "typeObj": "validateData", "tags": []}, {"busVersion": "1", "instance": "3749", "err": "0", "payload": {"data": {}, "dataId": "3749", "businessWorkflowInstanceId": "45776", "metricRef": "5689", "strategyRef": "7664"}, "typeObj": "validateData", "tags": []}, {"busVersion": "2", "instance": "sid-9C821864-D58C-4680-89F5-1C8E6CF57C50", "err": "0", "payload": {"data": [{"25", "29"}], "businessWorkflowInstanceId": "7941", "metricRef": "578a3367e4b0089faea19612"}, "strategyRef": "7810"}, {"typeObj": "validateData", "tags": []}]
5   ],
6   "resolvedAddress": "",
7   "originAddress": "",
8   "id": null
9 }
```

ESB: Notifiche e messaggi

- Notifica
 - Avviso nuova strategia
- Messaggio
 - Errore su una strategia

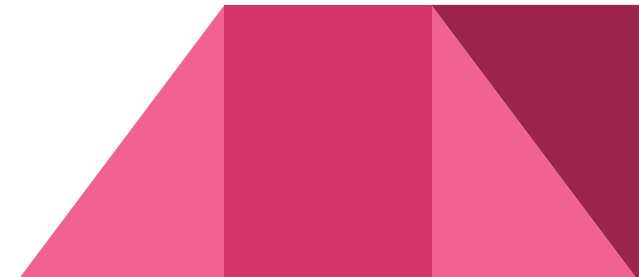


Notifica

<http://qips.sweng.uniroma2.it/gqm32/notify/eventonstrategy/>

```
{  
  "typeObj" : "base64-TerminalStrategy",  
  "operation": "...",  
  "instance": "...",  
  "phase": "phase1",  
  "busVersion": "..."  
}
```

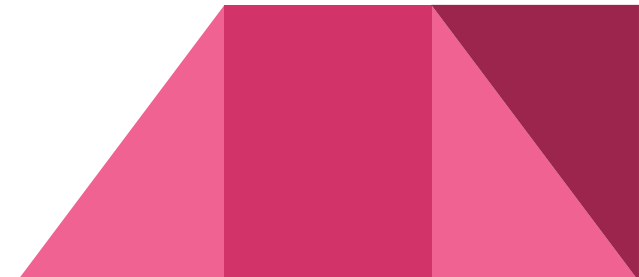
- Alla ricezione della notifica il sistema:
 - ~~Elimina le strategie che non sono più sul bus~~
 - Controlla la presenza di strategie nuove/modificate
 - Aggiorna le strategie alla versione più recente se necessario



Messaggi

Se viene inviato un messaggio di errore dalle fasi precedenti:

- Il sistema richiede all'utente se vuole modificare lo strategic plan o se vuole inviare una notifica alla fase1
- Se l'utente vuole inviare una notifica alla fase1 il sistema (attraverso ESB) invia una notifica del tipo
 - { "subject": " Alert on terminal strategy ", "body": "Please modify strategy with the id: 4" };



Tecnologie utilizzate

Sommario

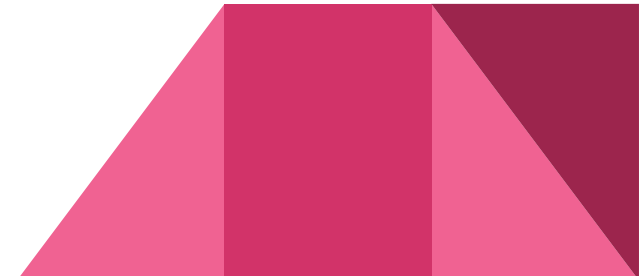
- Applicativo Back End

- Spring
- Activiti-Explorer

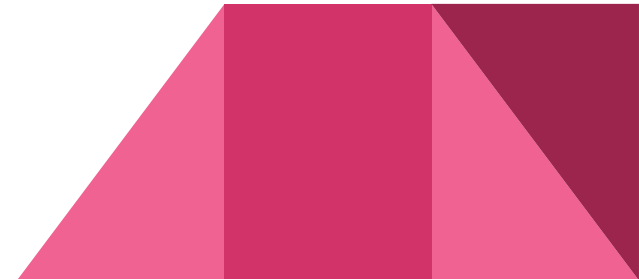
- Applicativo Front End

- Javascript - JQuery
- BootStrap
- Ajax
- HTML 5
- CSS

- Storage



- Enterprise Service BUS
- Sviluppo
 - Git
 - Eclipse
 - Firefox Firebug
 - Postman
- Progettazione, Documentazione e Comunicazione
 - Jira e Confluence
 - Slack



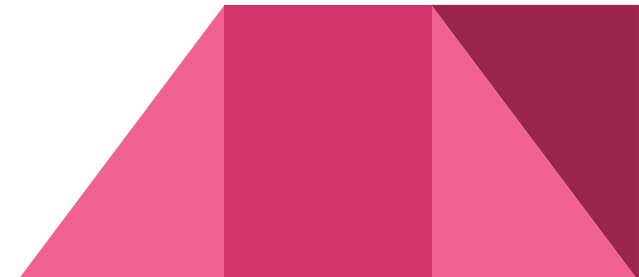
Applicativo Back End



GQM 3.2-4.2 è un'applicazione Web RESTful.

Il back end è sviluppato interamente in java.

Spring Application: viene utilizzato il framework open source Spring.



Activiti: Workflow Automation

Per gestire i workflow vengono utilizzate le web-app **Activiti-Explorer** e **Activiti-REST**.

Permette il disegno del modello di workflow mediante tool grafico e, tramite l'Engine Activiti offre un servizio di automatizzazione delle istanze dei processi.

- Funzionalità offerte al Workflow Developer
 - Creazione del modello di workflow
 - Modifica del modello di workflow
- Funzionalità offerte all' Execution manager
 - Avvio del processo
 - Stato avanzamento del processo
 - Terminazione del processo



In riferimento ai requisiti progettuali (fase 3.2 e 4.2) è stata fondamentale l'adozione di Activiti per usufruire dei suoi servizi di Workflow automation

Applicativo Front end



La view è sviluppata utilizzando HTML 5 e Javascript.

Libreria Javascript **JQuery** per implementare funzionalità **Ajax** e

creazione di elementi grafici come popup form mediante estensione

JQuery impromptu.

E' stato introdotto Ajax con lo scopo di effettuare richieste asincrone.

Scelta di JQuery:



- Inizialmente propensi all'utilizzo del framework AngularJS per integrazione utilizzo risorse JSON
- Poi, su consiglio dello scrum master, si è optato per l'utilizzo di JQuery per evitare il l'impiego di due controller, uno per il back end e uno per il front end

Bootstrap

Cos'è?



Bootstrap è una raccolta di elementi grafici, stilistici, di impaginazione e Javascript pronti all'uso per la creazione di applicazioni web.

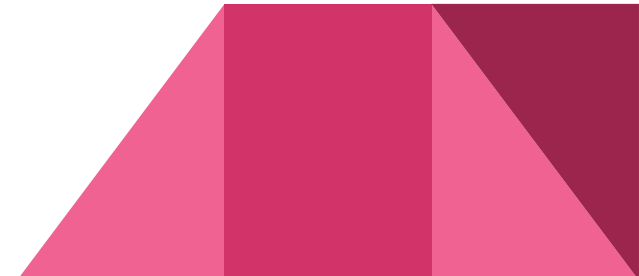
Come viene utilizzato?

In GQM 3.2-4.2 viene utilizzato il template **Joli admin** per impaginazione del codice HTML, creazione di elementi grafici per l'interazione con l'utente come ad esempio le form.

Bootstrap comprende quegli elementi CSS che permettono di definire la struttura e gli stili predefiniti per diversi elementi della pagina.



Non è stato necessario modificare stili.



Database

GQM3.2-4.2 utilizza un database Relazionale ed uno non relazionale.

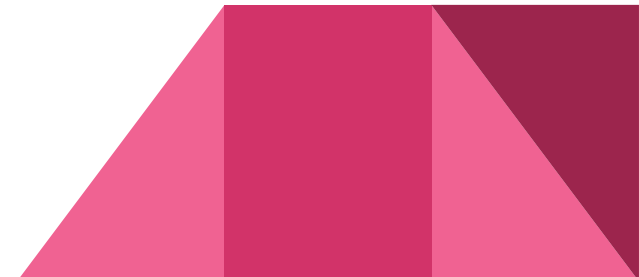
- DB Relazionale

- MySQL
- Database dell'applicazione Attività



- DB non relazionale

- MongoDB
- Database dell'applicazione GQM3.2-4.2
- Condiviso con la fase 3.1 e 4.1
- Orientato ai documenti
- Vengono memorizzati documenti JSON



La necessità di manipolare e gestire oggetti JSON é dovuta alla natura RESTful dell'applicazione.

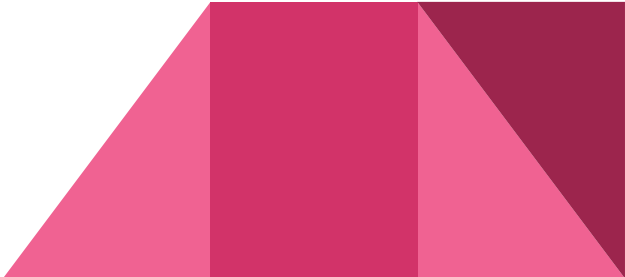
E' stato scelto MySQL (tra quelli supportati da Activiti) perchè:

- Gratuito
- Database relazionale veloce, semplice, performante

La scelta del DB noSQL è stato dettato da un precedente confronto tra i DB utilizzati più frequentemente oggi: MongoDB, Cloudant.

Ai nostri scopi la scelta sarebbe stata del tutto equivalente.

Tuttavia, si è scelto di utilizzare MongoDB perchè:

- Gratuito
 - Semplicità
 - Elevata documentazione in rete
- 

Ermes-Qip: Bus del sistema

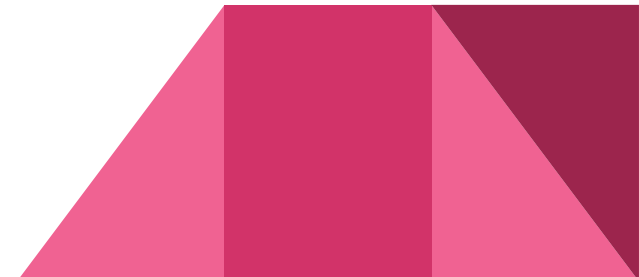
Applicazione java che permette la comunicazione tra le fasi GQM+S.

Cos'è un' Enterprise Service Bus?

Infrastruttura software per il supporto di Service-oriented architecture complesse.

Fornisce un middleware di comunicazione.

Può fornire servizi di coordinamento, sicurezza, messaggistica, instradamento intelligente e trasformazioni.



Ambienti di sviluppo

L'applicazione è stata sviluppata utilizzando l'ambiente Eclipse.



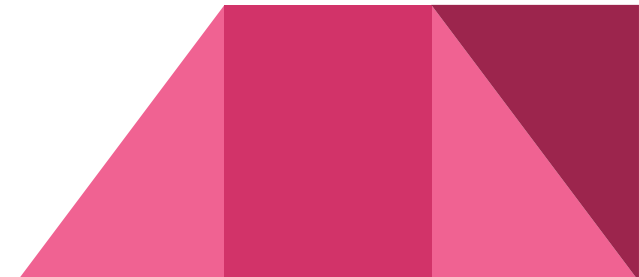
Git per la gestione dello sviluppo in team:

- open source software GitLab
- considerabile uno standard per il controllo versione distribuito



Utilizzo del tool Firebug per il debug applicativo.

Tool Postman per il test iniziale delle chiamate REST.



Progettazione, Documentazione e Comunicazione

Per la gestione operativa del progetto e la gestione della documentazione del software sono stati utilizzati i servizi offerti da Atlassian

- **JIRA** fornisce un ambiente per la gestione dei requisiti, tracciato dello stato del progetto, gestione dei processi, organizzazione del flusso di lavoro
- **Confluence** fornisce un ambiente per la gestione della documentazione del software e la gestione del Time sheet

Per la creazione della presentazione è stato utilizzato **google drive**, che ci ha permesso di creare una presentazione in cloud condivisa tra tutti i membri del team



E' stato utilizzato uno strumento per la comunicazione tra gli elementi del team,

Slack, software che offre:

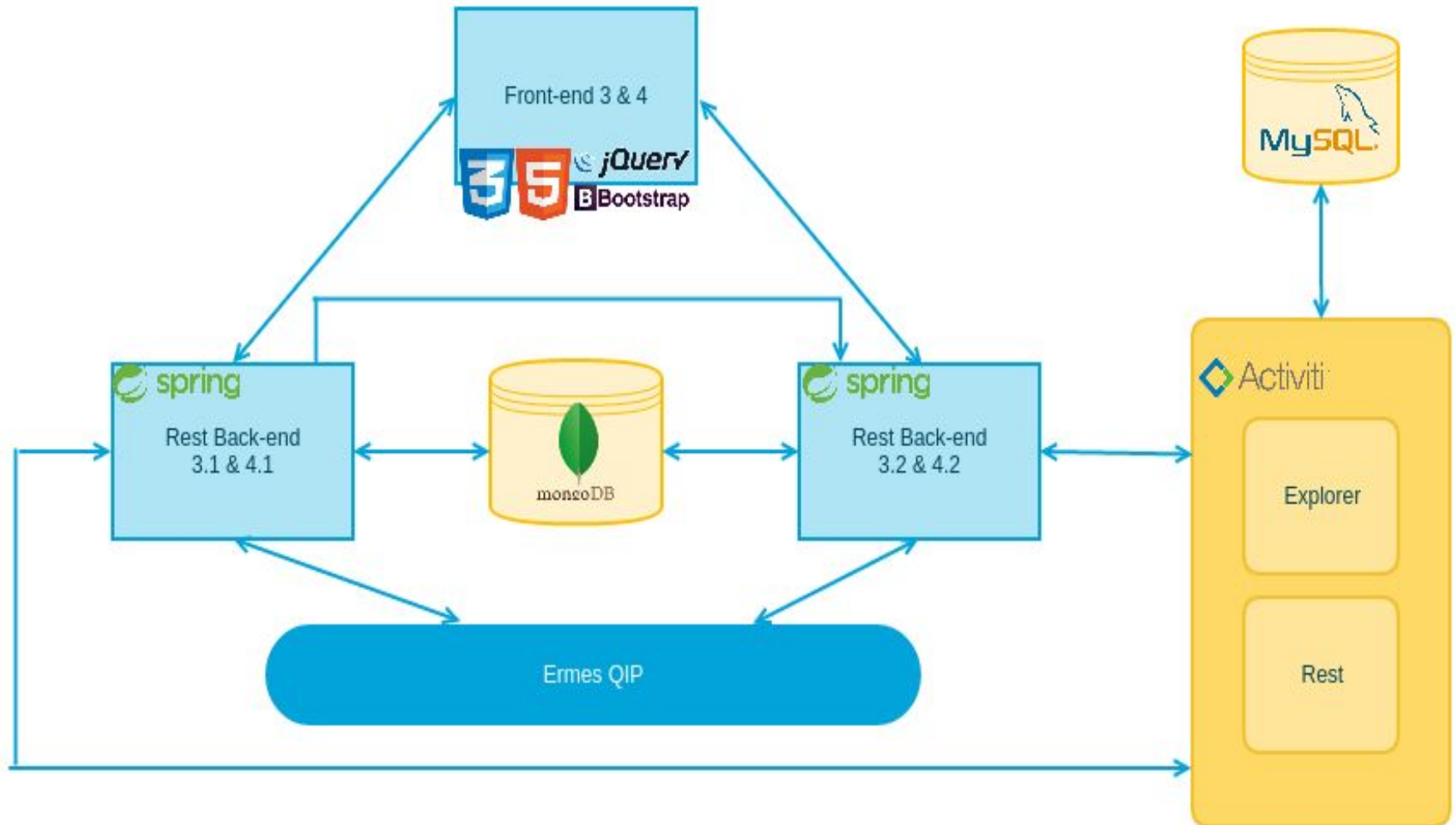
- chat di gruppo organizzate per tema e chat dirette
- Condivisione di documenti
- Condivisione di snippet di codice
- Integrazione con Git, offrendo la possibilità di merge requests a un membro del gruppo e la notifica di merge eseguiti.



Tutte le funzionalità di questo software sono state sfruttate per facilitare la comunicazione e coordinazione tra i membri del team

Architettura

Panoramica sull' architettura del sistema



Nel dettaglio (1)

- Applicazione totalmente RESTful
 - per un approccio più moderno
 - in linea con la logica dei servizi di rete (semplicità nell'uso del formato JSON)
 - Indipendenza dalla logica di presentazione
 - Maggiore manutenibilità ed estensibilità
- Web grafica in comune
 - per una maggiore sinergia dei due gruppi (coordinazione, gestione)
 - le view possono essere facilmente reindirizzabili tra utenti di differenti applicazioni
 - si può richiedere servizi REST ad entrambe le web app con estrema semplicità



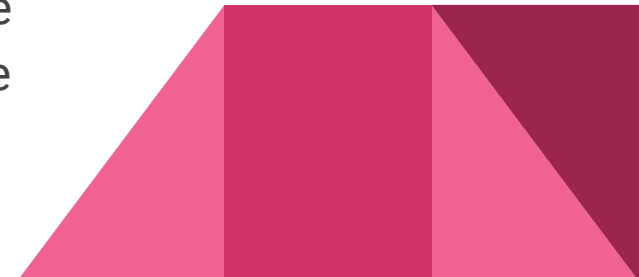
Nel dettaglio (2)

- **Integrazione completa con l'altra web app**
 - non solo un' esposizione di servizi, ma interazione dei servizi tra loro
 - comunicazione attraverso condivisione di memoria (MongoDB condiviso, classi condivise e concordate)
 - flusso di esecuzione generale dell' applicazione globale (due applicazioni RESTful più quella della grafica), precedentemente concordato
- **Interazione con il bus**
 - un flusso in entrata (lettura delle strategie dal bus)
 - un flusso in uscita (scrittura della lista dei CollectedData validati)
 - entrambi i servizi concordati con una forte integrazione e diversi incontri tra i gruppi



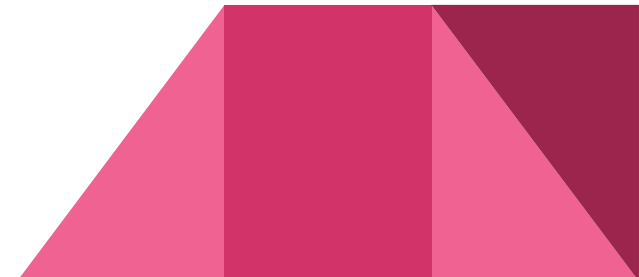
Nel dettaglio (3)

- **Utilizzo di Activiti-Explorer e Activiti-Rest**
 - utilizzo del Modeler per l'editing dei workflow
 - per sfruttare l' Engine di Activiti per la workflow automation dei workflow di business
 - per la gestione dei task ricorrenti di utenti dell' applicazione attraverso l' utilizzo di un workflow predefinito ("MetaWorkflow")
- **Persistenza garantita MongoDB**
 - perfettamente integrato con SpringBoot (Jpa, MongoRepository)
 - trovata una istanza free su mlab di 500mb (è stato possibile anche inizialmente condividere il database utizzando una istanza remota)
 - modificando l' application.properties è possibile cambiare le credenziali per l'autoconfigurazione (Cloudant troppo legato a IBM)

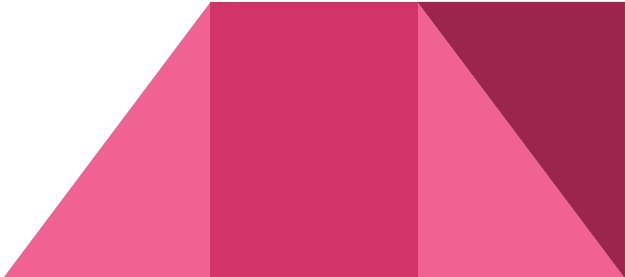


Lo use case generale

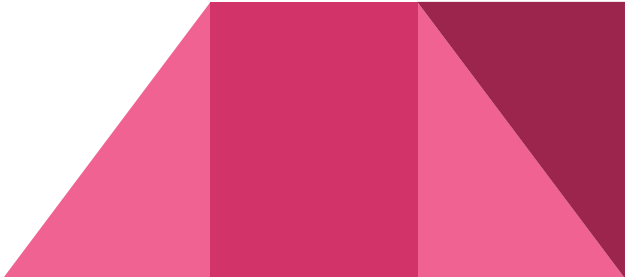
- E' il flusso completo dell'intera esecuzione
 - è stato concordato con il gruppo 3.1 e 4.1
 - viene gestito da ambedue le webapp
 - trova il suo cuore all'interno del "MetaWorkflow" che gestisce la fase 3 di GQM+Strategies
- Sostanzialmente si compone di 3 fasi
 - creazione di piani strategici (dalle strategie che arrivano dal bus)
 - esecuzione di un Metaworkflow associato ad una strategia (gestendo l'iter ricorrente e gli utenti in esso coinvolti)
 - avvio del workflow di business editato e attrezzato nell'iter, e si mette in attesa di feedback dal workflow di business, dove avviene la fase di raccolta dati e validazione.



Due parole sul Metaworkflow

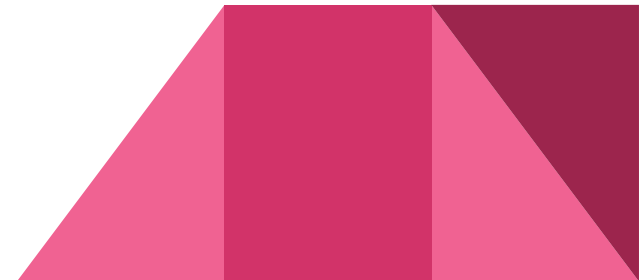
- E' gestito dalla web application 3.1 e 4.1
 - E' presente al suo interno
 - Viene associato da uno Strategic Planner ad una strategia dello Strategic Plan, e da lì deployato e avviato
 - Fasi principali del MetaWorkflow
 - Strategic Planner per il Workflow developer (1° User Task)
 - Editing, deploy ed export del Workflow di business (2°, 3° e 4° User Task)
 - Pianificazione delle misure (5° User Task)
 - Pianificazione della validazione (6° User Task)
 - GQM Expert (Execution Manager) avvia il Workflow di business (7° User Task)
- 

User Tasks del Metaworkflow

- User Tasks prima sono unassigned e assegnati ad un candidate user
 - L'utente facente parte di quel group in Attività fa il relativo claim del task
 - L'utente visualizza tasks unassigned, e se li assegna. Da quel momento in poi gestisce lui il task
 - Finito il lavoro su quel task lo completa e il flusso di quel Metaworkflow va avanti
 - Rendono possibile la pianificazione operata nella fase 3 di GQM⁺Strategies
- 

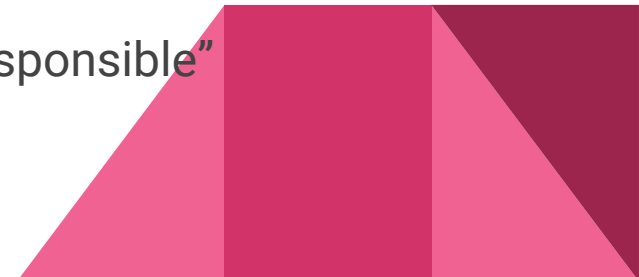
Workflow di business

- Workflow che si propone di conseguire la strategia a cui è associato
- La relazione è uno ad uno tra strategia e workflow
- All' interno sono disposti User Tasks generici che coinvolgono uno o più utenti di GQM⁺Strategies
- Esistono User Task “speciali”, assegnati ad un particolare [candidate group] che inserisce un dato, e il task viene successivamente riassegnato ad un Validator, che deve validare il dato



User Task di Misura e Validazione (fase 4)

- User Task che deve gestire la raccolta di un dato, e il processo di validazione per quel dato
- Perciò è uno User Task che non deve essere completato se non quando è effettivamente validata la misura
- Coinvolge un attore della nostra applicazione e un [attore dell 'intera organizzazione] che va ad inserire la misura
- Si è gestito tale processo attraverso delle variabili di stato (variabile "state")
 - "1": deve essere inserita la misura
 - "2": deve essere validata la misura
 - "3": errore nella misura, e la parola ritorna al "responsible"



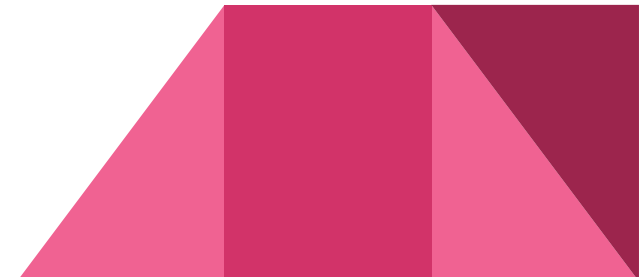
Evoluzione del progetto

Un progetto non lineare!

Progetto non è frutto di uno sviluppo lineare, in cascata ma **risultato di una evoluzione continua**

Le 4 tappe cronologiche di questo lungo viaggio:

1. sviluppo individuale del progetto (con incontri del martedì)
2. sviluppo attraverso architettura concordata
3. cambio di architettura
4. forte integrazione finale con front end unificato



Prima fase del progetto

I gruppi 3.1/4.1 e 3.2/4.2 hanno iniziato regolarmente ad incontrarsi di martedì:

- approfondendo aspetti teorici;
- elaborando la loro documentazione;
- affacciandosi al nuovo mondo della workflow automation;

*Non si pensava ancora di sviluppare in futuro due applicazioni tra loro così **fortemente** integrate ed interagenti.*

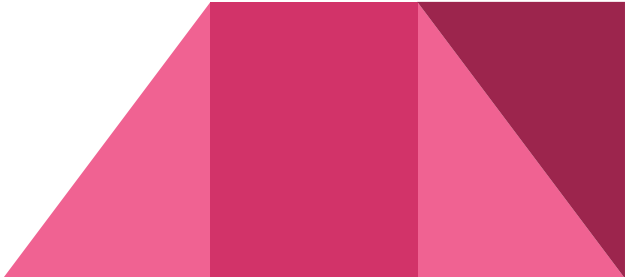


Seconda fase del progetto

Inizio di una collaborazione molto fitta tra i due gruppi

- nascita dell' idea del **Metaworkflow**;
- idea di **customizzare** Activiti-Explorer e il suo Modeler;
- inizio della progettazione di una parte dell'applicazione per conto proprio;
- indirizzare un utente dalla propria web application ad Activiti-Explorer (basandosi sulla suddivisione che fa Activiti dei ruoli)

La customizzazione prevedeva la possibilità per un Workflow Developer di inserire:

- User Task di misura
 - Service Task per la validazione
- 

Problematiche di questo approccio

Si è abbandonata questa scelta per diversi motivi:

- non si era riusciti a customizzare in maniera desiderata Activiti-Explorer;
- i permessi degli utenti sono gestiti male su Activiti, nel senso che tutti possono fare tutto

In particolar modo il secondo punto non consentiva, ad esempio, di distinguere adeguatamente un esecutore di un task da un workflow developer.



Terza fase del progetto

Utilizzo di Activiti-Rest:

- esternamente alla nostra applicazione;
- ricostruendo tutto ciò che è necessario di Activiti-Explorer nella nostra web application

Vantaggi:

- più flessibilità;
- utilizzo solo di ciò che è strettamente necessario;
- piena risoluzione delle problematiche riscontrate precedentemente

Conseguenze:

Non è più possibile accedere direttamente al Modeler di Activiti!



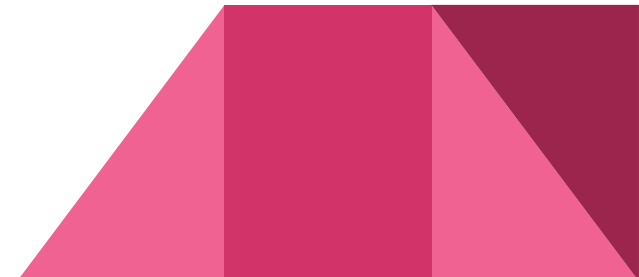
Il problema del Modeler di Activiti (1)

Activiti-Explorer presenta un Modeler integrato, che si è tentato di estrarre ma senza successo..

Idea: bypassare il problema, viste le problematiche del cambio di architettura, con una redirectione del tipo:

<http://localhost:8080/activiti-explorer/modeler.html?modelId=139167>

noto il modelId di un modello, creato attraverso una chiamata Rest ad Activiti



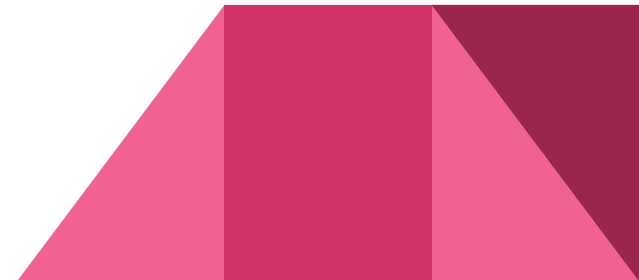
Il problema del Modeler di Activiti (2)

Ma Activiti-Explorer presenta una gestione della sessione utente => bisogna prima effettuare una autenticazione

Le strade percorse:

- ricerca di qualche servizio rest per fare il login (online nei forum veniva suggerito es. <http://localhost:8080/activiti-rest/service/login>)
- si è provato a fare basic authentication come per Activiti-Rest

Tutte queste soluzioni hanno comportato un fallimento.

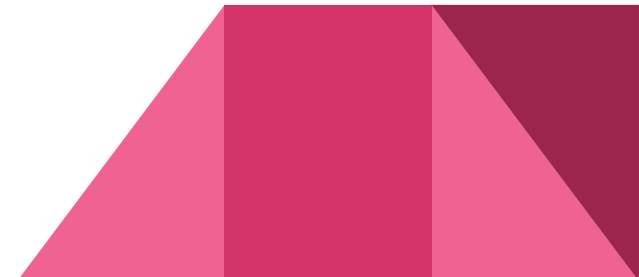


Il problema del Modeler di Activiti (3)

Si è scelto pertanto di loggare l'utente aprendo una nuova pagina al seguente link:

[http://localhost:8080/activiti-explorer/?username="+username+"&password="+password](http://localhost:8080/activiti-explorer/?username=)

Dopo di che si chiude questa pagina e si redirige l'utente al link del modeler, per editare il workflow.



La terza fase nel dettaglio

Da un punto di vista logico e architetturale:

- E' stato concordato il Metaworkflow con l'altro gruppo
- Si è concordata l'intera architettura dell'applicazione
- Si è stabilito il flusso logico
- Si è unificata la documentazione

Da un punto di vista pratico:

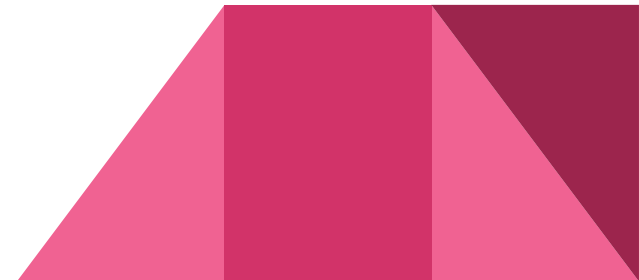
- Sono state sviluppate diverse User Stories (con relativo frontend)
- Si è iniziato a dialogare attraverso dei servizi Rest con Activiti



Quarta fase del progetto

Per migliorare le sinergie di entrambi i gruppi:

- sono state realizzate due web app RESTful, una per ogni gruppo
- una web application in comune contenente il frontend
- sono stati concordati dei servizi Rest in comune
- si sono condivise le entity, poiché si è scelto di far comunicare le due web application per memoria condivisa (MongoDB) e richiedendo i servizi ad entrambe le web app, direttamente dalle View (in realtà loro richiedono dei servizi attraverso comunicazione Rest-template alla nostra web-app per rielaborare i dati per la gestione del riassegnamento dei task)



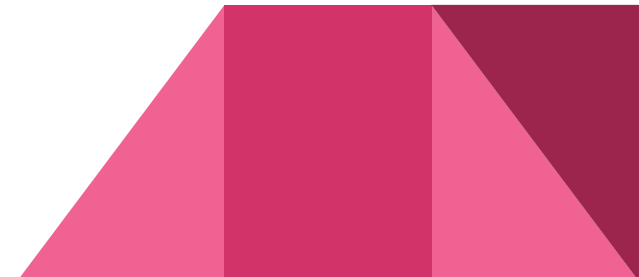


Activiti

Activiti

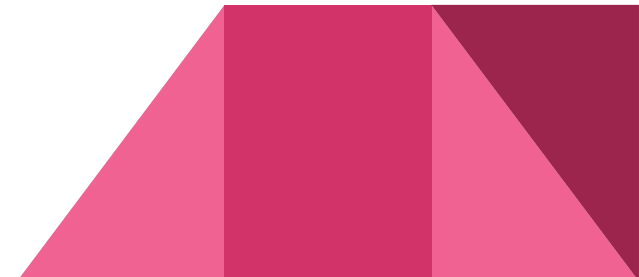


- Cos'è Activiti?
 - Workflow Engine BPMN 2.0
 - Naturale evoluzione di jBPM
 - Gestisce l'esecuzione automatica di processi
 - Gestisce la persistenza dello stato dell'esecuzione
- Vantaggi:
 - Semplifica la collaborazione tra Business e IT
 - Alta integrazione (database, Spring, REST)
 - Semplicità

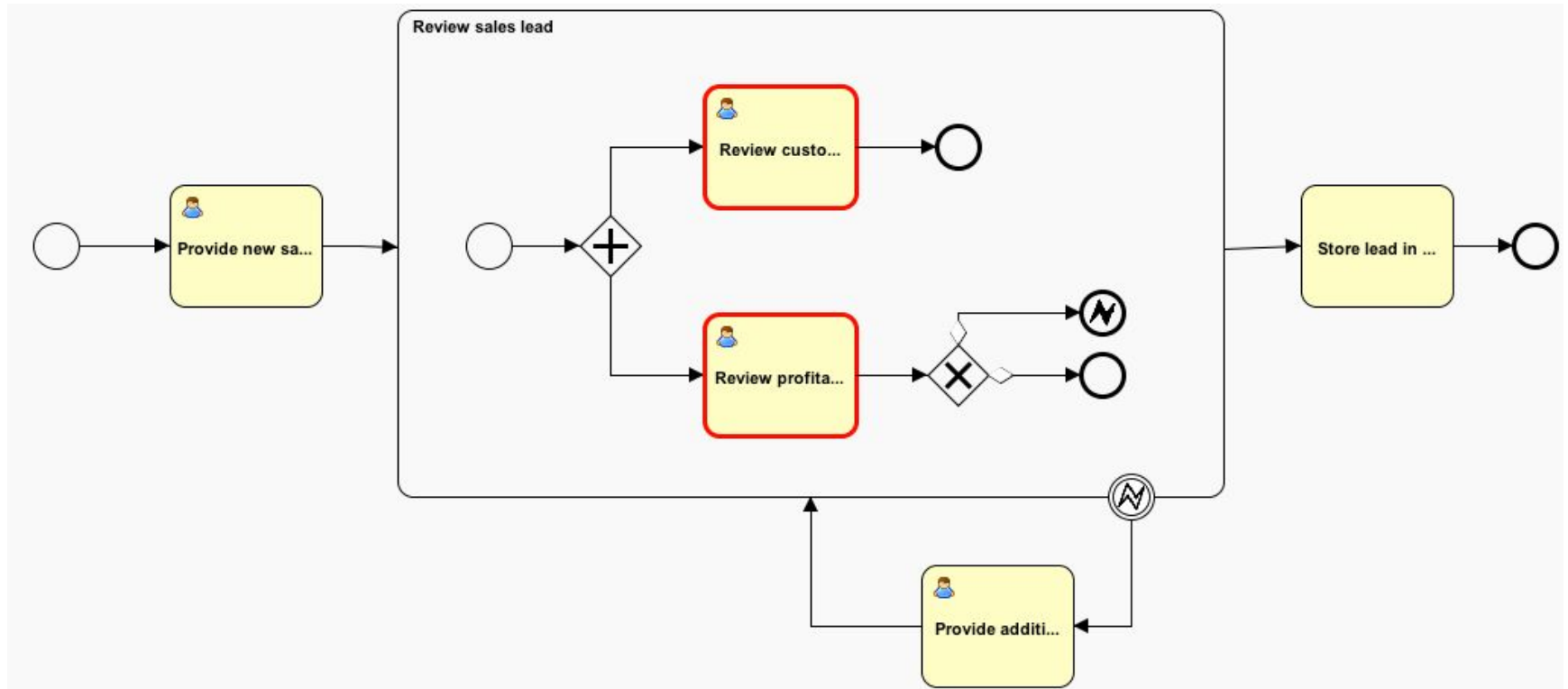


Activiti - Suite

- Suite di applicazioni:
 - **Modeler**: interfaccia web di editing
 - **Designer**: plugin per Eclipse
 - **Engine**: processore di workflow
 - **Explorer**: interfaccia web per deploy, avvio ed esecuzione dei workflow
 - **Cycle**: webapp per collaborazione tra utenti di business e sviluppatori software

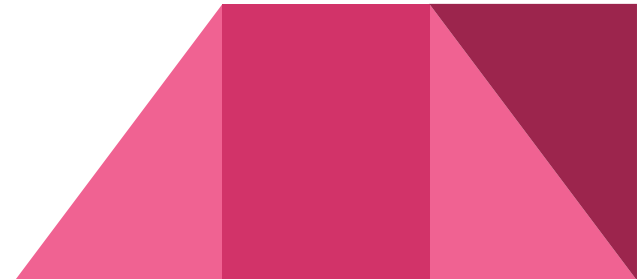


Activiti - Esempio di Workflow



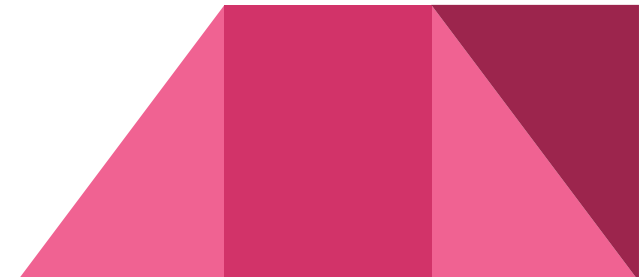
Attività - Componenti

- Process
- Task
 - User Task
 - Gestione dei form
 - Service Task
 - Diverse implementazioni
- Start/end event
 - Gestione degli errori
- Gateway
 - Exclusive, inclusive, parallel
- Subprocess
- Variabili



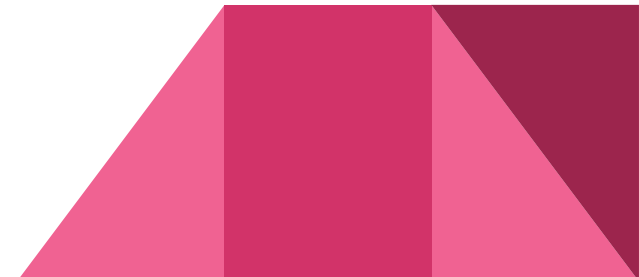
Activiti & GQM⁺Strategies

- Idea iniziale: creare nuove entità nel modeler al di sopra del workflow engine
 - Nessuna modifica al workflow engine: una volta definite nell'Activiti modeler custom, possono essere eseguite su un qualsiasi workflow engine BPMN 2.0-compliant.
- Modifica semplice da effettuare nel plugin Eclipse (Service Task estensibili a piacimento), molto più difficile nel modeler, non progettato per essere esteso a questo livello. Un supporto per l'estensione è stato previsto per la prossima versione di Activiti Modeler.



Activiti & GQM⁺Strategies - Nuove entità

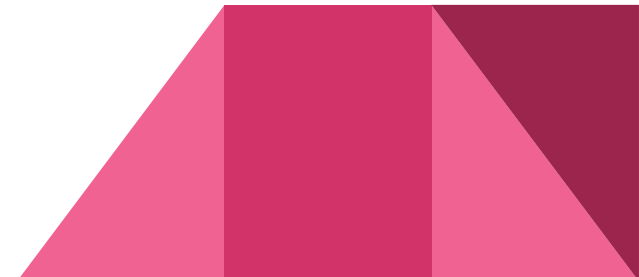
- **Validation Service Task**
 - Valida i dati in modo automatico: anziché specificare una classe Java da eseguire, si specifica l'operazione di validazione da effettuare
- **Validation User Task**
 - Permette all'utente di supervisionare la validazione dei dati
- **Validation End Error Event**
 - Termina il processo specificando che devono essere ridefinite le misure
- **Validation Exclusive Gateway**
 - Controlla che la validazione sia andata a buon fine, e gestisce il flusso di esecuzione





Activiti & GQM⁺Strategies - Problematiche

L'approccio in seguito è stato abbandonato, in favore di un approccio RESTful che implementa la logica di validazione al di fuori di Activiti. Tra le motivazioni troviamo:

- Modeler non estensibile facilmente
- Implementazione dipendente da un'applicazione esterna
- Impossibilità di aggiornamento
- Disomogeneità dell'interfaccia utente












Activiti & GQM⁺Strategies - Approccio RESTful


 Task Execution - 

#8 My task

Task description

var1		
var2		Value 2
var3		2016-04-13
		12:44:00
var4	<input type="checkbox"/>	True
var5		
var6		questo campo è required
var7		Value 2
var8		2016-04-13
		12:44
var9	<input checked="" type="checkbox"/>	false
var10		

Activiti & GQM⁺Strategies - Approccio RESTful



Workflow Definition -

Strategic Plans

Show entries Search:

Strategic plans

- Monitorare e valutare corsi di studio

Terminal Strategies

Show entries Search:

- Implementare valutazione soddisfazione
- Monitorare votazione esame

Showing 1 to 2 of 2 entries

Previous **1** Next

Strategy description

Gestire lo svolgimento degli esami mediante la registrazione delle votazioni degli studenti.

Create Workflow

Activiti-REST

Activiti-Rest (1)

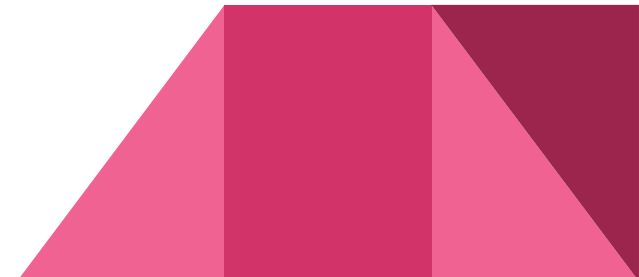
Activiti include REST API verso l'Activiti Engine attraverso il war di activiti-rest che può essere deployato su un servlet container come Apache Tomcat (dialogando con Activiti-Explorer attraverso API REST)

NON E' L'UNICA SOLUZIONE

Infatti può essere utilizzata da un'altra web application, includendo la servlet e il suo mapping nell'applicazione, e tutte le dipendenze ad activiti-rest nel classpath.

Una delle alternative era infatti quella di **includere** Activiti e il suo **process Engine** nella nostra web app.

Noi abbiamo deciso di far dialogare la nostra web app con l'engine di Activiti-Explorer attraverso Activiti-Rest.



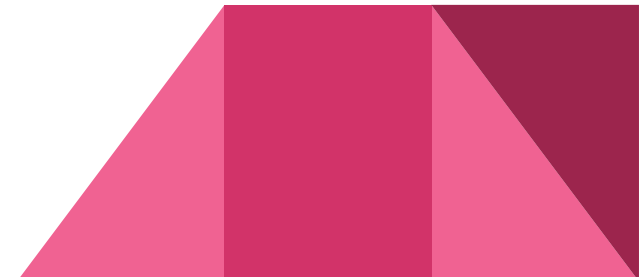
Activiti-Rest (2)

Di default l' Activiti Engine utilizza il database H2 embedded.

L'applicazione presenta dei file SQL che al lancio dell'applicazione popolano il DB, ad esempio con utenti di default.

Per poter far dialogare Activiti-Rest con Activiti-Explorer quello che in genere si fa è settare nel file `db.properties` le credenziali di accesso a MySQL o un altro database relazionale supportato, di entrambe le web app.

(Nel caso di MySQL si è riscontrato il problema dell' assenza dei connettori, che sono stati aggiunti senza problemi.)

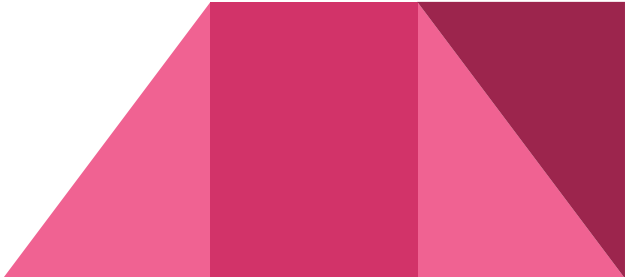


Activiti-Rest (3)

Tutte le chiamate REST ad Activiti richiedono una HTTP Basic authentication per l'accesso, pertanto l'url di richiesta generalmente utilizzato è :

<http://username:password@localhost...>

Tra gli utenti di Activiti-Explorer è presente l'utente "kermit", che ne è anche l'amministratore. Di conseguenza in accordo a quanto stabilito con il gruppo della fase 3.1 e 4.1, si è deciso di operare la Basic Authentication attraverso le credenziali di "kermit", settandole nel file application.properties dell'applicazione.

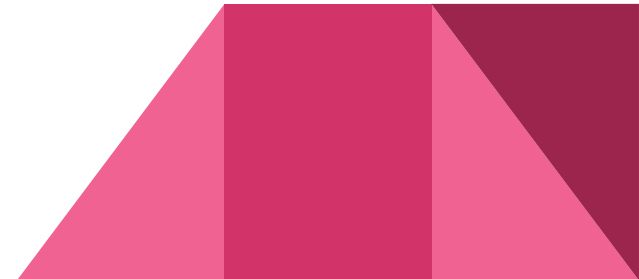


I servizi Rest maggiormente impiegati

Activiti-Rest espone praticamente quasi tutte le chiamate necessarie per customizzare una applicazione web facendole integrare le funzionalità di Activiti, sia con che senza Engine incorporato che senza (utilizzando Activiti-Explorer).

Nella nostra applicazione sono 3 le tipologie di chiamate con cui si è particolarmente interagito:

- task
- form
- process



Activiti: esempio di chiamata Rest

GET <http://kermit:kermit@localhost:8080/activiti-rest/service/runtime/tasks?assignee=tigro>

```
{
  "data": [
    {
      "id": "3523514",
      "url": "http://localhost:8080/activiti-rest/service/runtime/tasks/3523514",
      "owner": null,
      "assignee": "tigro",
      "delegationState": null,
      "name": "userTaskA",
      "description": null,
      "createTime": "2016-07-17T00:41:26.000+02:00",
      "dueDate": null,
      "priority": 50,
      "suspended": false,
      "taskDefinitionKey": "userTaskA",
      "tenantId": "",
      "category": null,
      "formKey": null,
      "parentTaskId": null,
      "parentTaskUrl": null,
      "executionId": "3523510",
      "executionUrl": "http://localhost:8080/activiti-rest/service/runtime/executions/3523510",
      "processInstanceId": "3523510",
      "processInstanceUrl": "http://localhost:8080/activiti-rest/service/runtime/process-instances/3523510",
      "processDefinitionId": "process:14:3480923",
      "processDefinitionUrl": "http://localhost:8080/activiti-rest/service/repository/process-definitions/process:14:3480923",
      "variables": []
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "id",
  "order": "asc",
  "size": 1
}
```

id di runtime del task

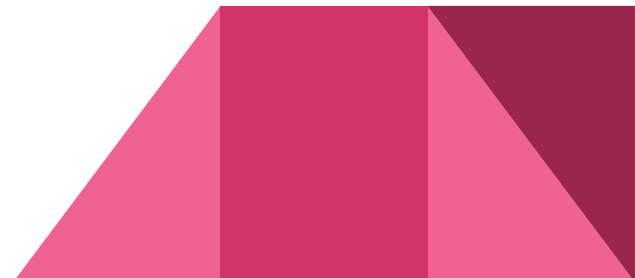
id di deploy del task

Paginazione che fa Activiti

E' presente un solo User Task in Activiti, assegnato all'utente tigro, che si trova in un processo in esecuzione, che deve essere completato

Chiamate esposte al gruppo 3.1 e 4.1

- POST `activiti/suspendTask/{taskId}`
 - Richiamata quando si fa il **submit** di un form per lo storage del form inserito da un utente
- POST `activiti/completeUserTask/{taskId}`
 - Per effettuare il **complete** di uno User Task generico
 - Per effettuare il **riassegnamento** di un MeasureTask creando delle variabili locali (tra cui `state = "1"` e `assignee = ""`)



Validazione

Pianificazione della validazione dei dati

- Prima che i dati misurati siano memorizzati e analizzati è necessario avviare un processo di controllo della qualità dei dati raccolti.
- Tale controllo viene pianificato e eseguito dal validator.
- La pianificazione del controllo della qualità dei dati raccolti dovrebbe prevedere una “checklist” di possibili problemi di integrità dei dati e delle procedure di analisi dei dati appropriate.



Problemi di integrità dei dati

Completeness:

- Gestire i dati mancanti.
- Il dato può mancare o essere del formato non corretto.

Contromisure:

- Ripetere la misura.
- Derivare la misura dai dati precedentemente raccolti. Ad esempio si può inserire la media dei valori precedenti.



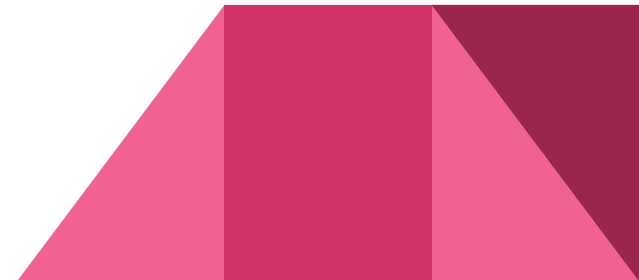
Problemi di integrità dei dati

Abnormal Data:

- Il dato misurato rappresenta un outlier.

Contromisure:

- Raccogliere nuovamente il dato fuori scala e "correggilo".
- Escludere il dato fuori scala

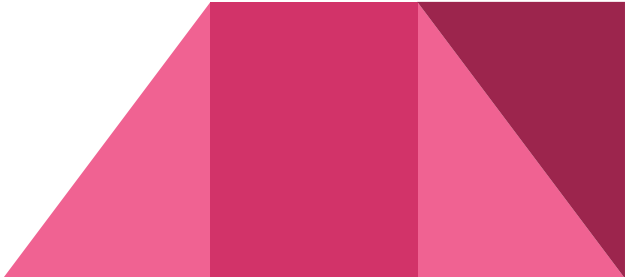


Problemi di integrità dei dati

Internal Consistency:

- Verificare la relazione tra i dati relativi allo stesso attributo dello stesso oggetto.

Contromisure:

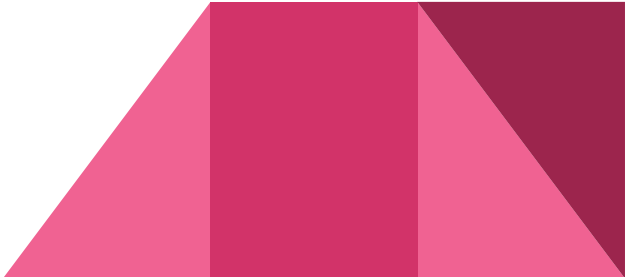
- Verificare se i dati sono stati raccolti in base alle metriche definite.
 - Ripetere la misura (a seguito della verifica definita nel punto precedente).
- 

Problemi di integrità dei dati

Plausibility:

- Verificare la relazione tra i dati relativi a diversi attributi dello stesso oggetto.

Contromisure:

- Verificare se i dati sono stati raccolti in base alle metriche definite.
 - Ripetere la misura (a seguito della verifica definita nel punto precedente).
- 

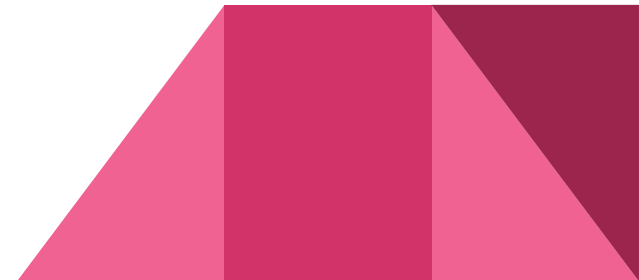
Problemi di integrità dei dati

External Consistency:

- Verificare la relazione tra i dati raccolti e il mondo esterno (fatti o assunzioni).

Contromisure:

- Ripetere la misura.



La validazione nel nostro progetto...

- Non gestisce i problemi di completeness, in quanto sono stati gestiti durante la fase di raccolta dei dati.
- Consente di risolvere i problemi relativi agli altri problemi di integrità dei dati.
- Permette di effettuare il confronto tra due valori numerici.
 - Il primo valore è il dato che si sta validando o un valore ottenuto dalla storia dei dati raccolti per quella misura.
 - Il secondo valore può essere un valore ottenuto da alcune misure, oppure un valore inserito dal validator.

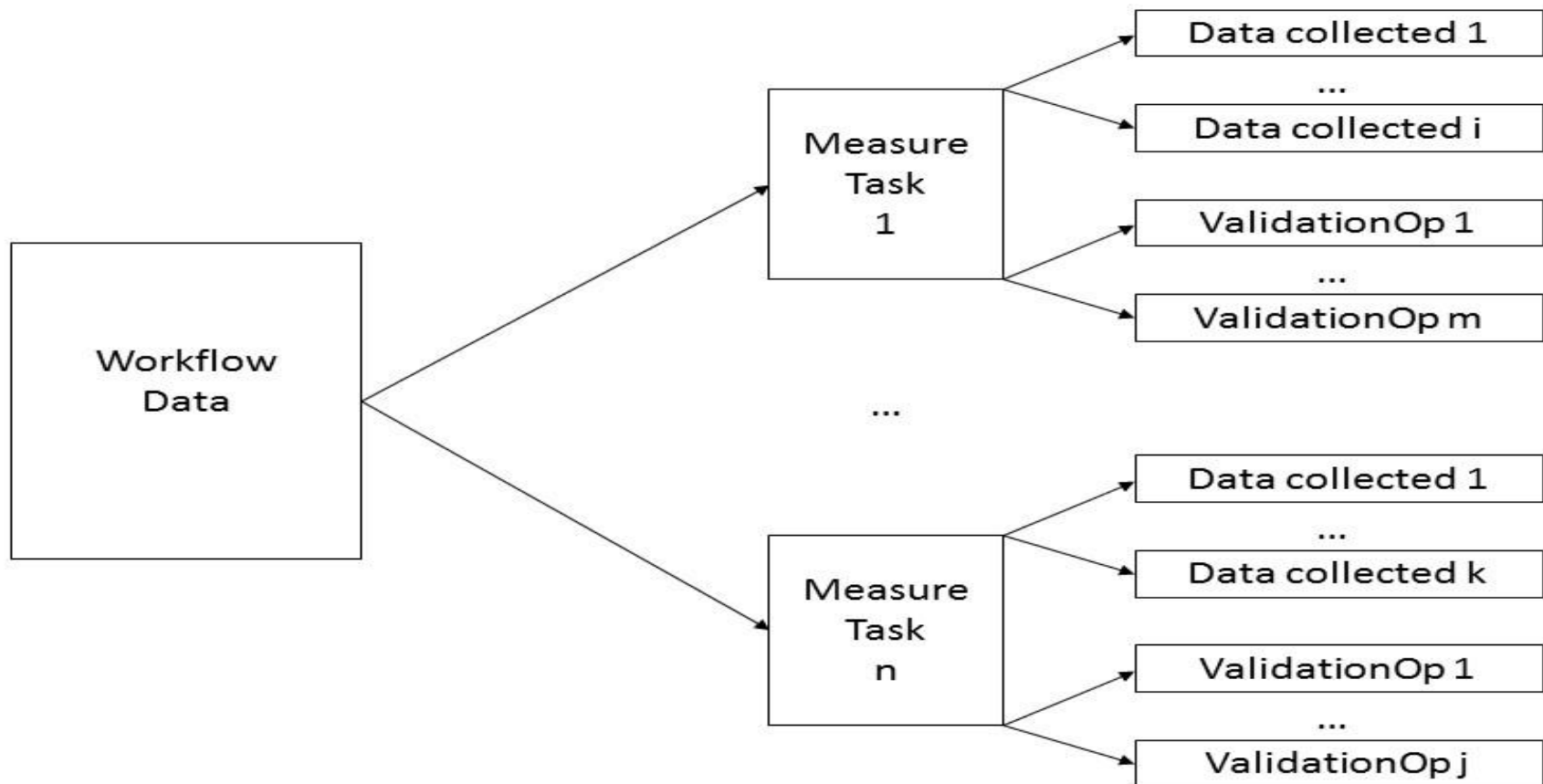


La validazione nel nostro progetto...

- Un valore può essere ricavato da un set di dati collezionati nel tempo e associati ad una misura.
 - Per questo motivo sono state previste operazioni come la somma, la media e la varianza.
- I due valori devono essere confrontati tra loro.
 - Per tale motivo sono state previste operazioni di confronto come $=$, $<$, $>$, inside e outside interval.
- Se la validazione non ha successo si deve poter eseguire una contromisura.
 - Ripetere la misura, Controllare la metrica o ignorare la validazione.

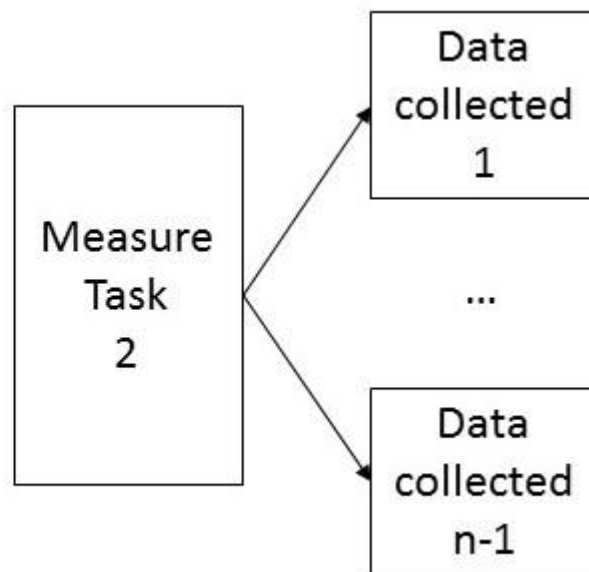
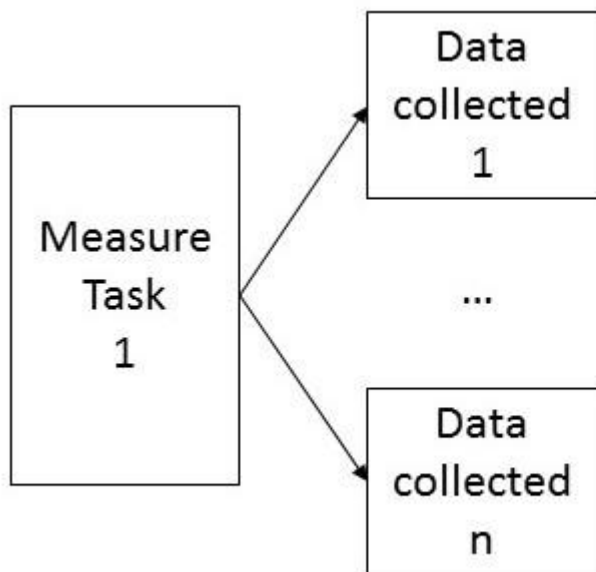


Per capire meglio la validazione



Per capire meglio la validazione

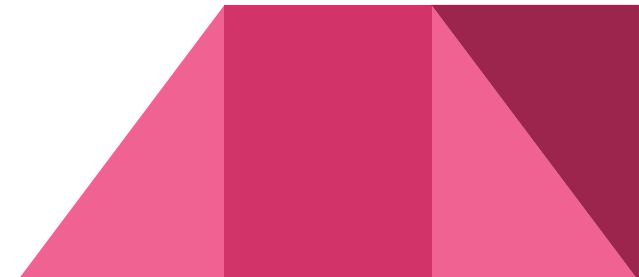
Potrei voler validare il dato raccolto nel Measure Task 1, confrontandolo con i dati raccolti nel Measure Task 2.



Pianificazione della validazione (Fase 3)

Il validator per effettuare la pianificazione dovrà eseguire i seguenti passi:

- Selezionare un task di misura per cui definire un piano di validazione.
- Aggiungere, modificare o eliminare delle operazioni di validazione.
- Definire tutti i dettagli relativi ad un'operazione di validazione.



Pianificazione della validazione (Fase 3)



Measure List

Task Name	Metric Name	Description Name
voto esame analisi	Metrica di prova	Descrizione di prova
voto esame geometria	Metrica di prova	Descrizione di prova
voto esame fisica	Metrica di prova	Descrizione di prova

Complete planning

Pianificazione della validazione (Fase 3)

Validation Operations List - voto esame analisi

Name	Description	Type	Actions
Voto valido	Voto deve essere compreso tra 18 e 30	INTERNAL_CONSISTENCY	 

Add

Save

Pianificazione della validazione (Fase 3)

Define validation - New Validation

Insert name and choose the validation type

Name



Voto valido

Insert the name of validation

Description



Voto deve essere compreso tra 18 e 30

Insert the description of validation

Validation Type

Internal Consistency



Select the type of validation

Measured Data

Cardinality

One



Select the cardinality of the measured data.

Validation criteria

Comparison type

Inside interval



Select the type of comparison.

Input Measure

User defined



Select if the second measure is inserted by the user or is an other measure collected by the system.

Value type

Interval value



Select the type of value to insert.

Insert Interval



18



30



Insert a interval to execute the comparison

Countermeasures

☒ Repeat Measure

☐ Control metrics

☒ Ignore validation

Clear Form

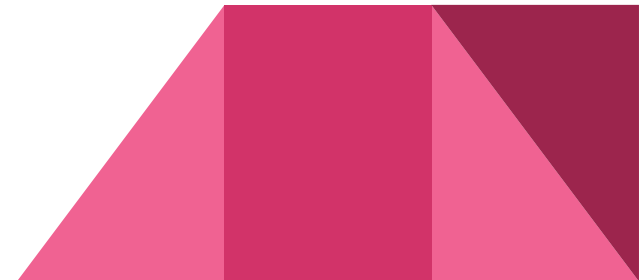
Submit

Esecuzione della validazione (Fase 4)

Il validator per eseguire i piani di validazione in fase di esecuzione dovrà eseguire i seguenti passi:

- Selezionare un'operazione di validazione.
- Eseguire quell'operazione di validazione.
- Verificarne l'esito.

Per riuscire a validare un dato tutte le operazioni di validazione associate ad una misura dovranno avere successo.



Esecuzione della validazione (Fase 4)

Execute validation - Voto valido

Validation Description

Voto deve essere compreso tra 18 e 30

Execute Validation

Analisi e Progettazione

Attori



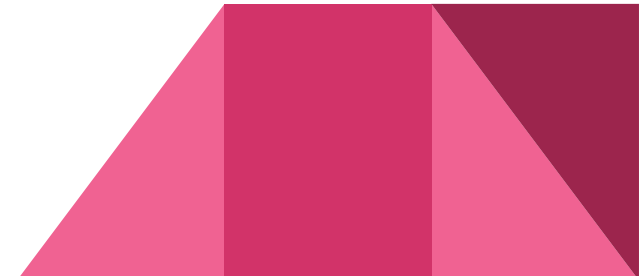
Strategic
Planner

L'attore che nella fase 3 definisce gli strategic plans.



Workflow developer

L'attore che nella fase 3 crea o modifica un modello di workflow.





Validator

L'attore che nella fase 3 pianifica la validazione dei dati e la gestisce nella fase 4.



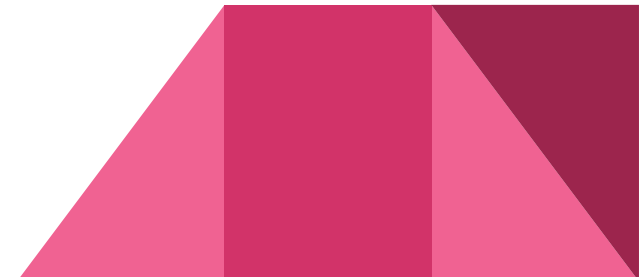
Execution Manager

L'attore che nella fase 4 esegue il modello di workflow.



Task Executor

L'attore a cui viene assegnato un task del workflow nella fase 3 e che lo esegue nella fase 4.





Enterprise bus
Ermes QIP

Il bus del sistema che interagisce attivamente con l'applicazione.



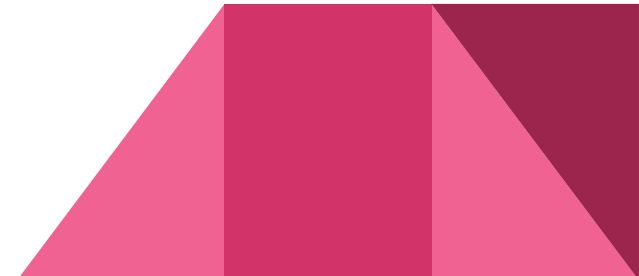
Activiti Explorer

L'interfaccia grafica con il workflow engine.



Activiti REST

L'interfaccia REST con il workflow engine.



Stakeholders



Project Manager

E' il principale committente dell'applicazione.



Strategic
Planner

Utente che nella fase 3 definisce gli strategic plans.





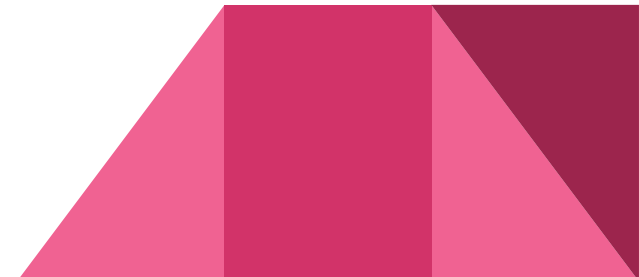
Workflow developer

Utente che nella fase 3 crea o modifica un modello di workflow.



Validator

Utente che nella fase 3 pianifica la validazione dei dati e la gestisce nella fase 4.





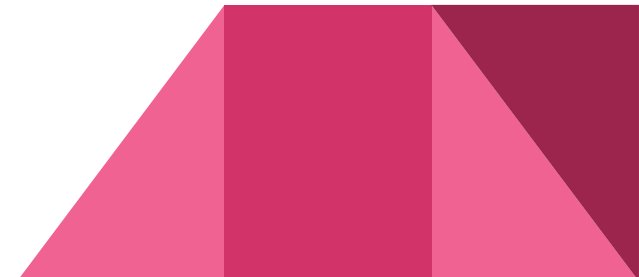
Execution Manager

Utente che nella fase 4 esegue il modello di workflow.




Task Executor

Utente a cui viene assegnato un task del workflow nella fase 3 e che lo esegue nella fase 4.

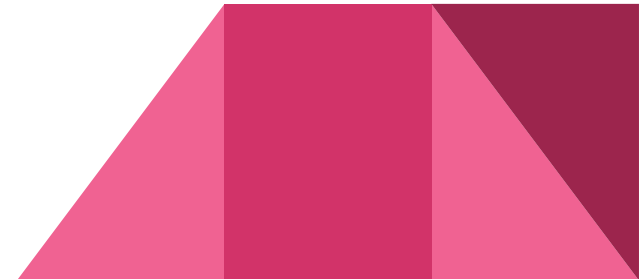


Stakeholder Request

- REQ1: lo Strategic Planner, nella fase 3, vuole dettagliare gli strategic plans per permettere al workflow developer di realizzare il workflow.
 - REQ2: Il Workflow Developer, nella fase 3, vuole creare o modificare un modello di workflow.
 - REQ3: Il Validator, nella fase 3, vuole pianificare la validazione dei dati.
 - REQ4: L'Execution Manager, nella fase 4, vuole eseguire un modello di workflow.
 - REQ5: Il Validator, nella fase 4, vuole gestire la validazione dei dati.
 - REQ6: Il Project Manager vuole ottimizzare l'esecuzione di GQM+Strategies all'interno dell'azienda.
 - REQ7: Il Workflow Developer vuole creare e modificare il workflow in modo semplice.
 - REQ8: Il Project Manager vuole che il sistema interagisca con i sistemi preesistenti.
- 

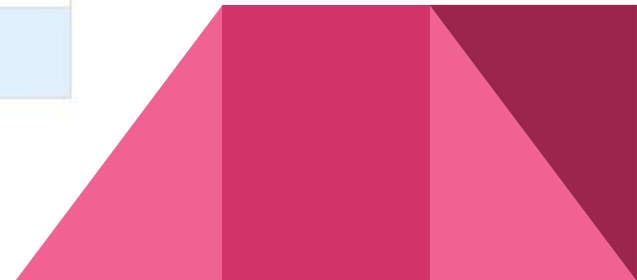
Features

- FEAT1: Definizione dettagliata degli strategic plans.
- FEAT2: Realizzazione e modifica del workflow tramite tool grafico.
- FEAT3: Esecuzione automatica del workflow.
- FEAT4: Pianificazione della validazione dei dati.
- FEAT5: Validazione semi automatizzata dei dati.
- FEAT6: Integrazione con i sistemi preesistenti tramite il bus.



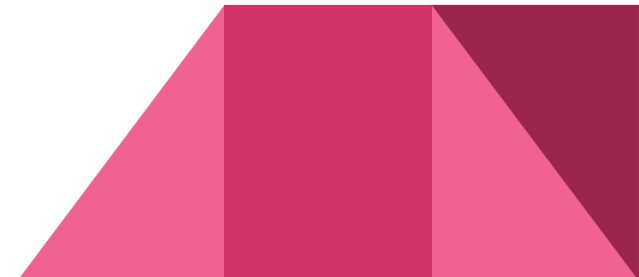
Mappatura da stakeholder Request a Features

	FEAT1	FEAT2	FEAT3	FEAT4	FEAT5	FEAT6
REQ1	*					
REQ2		*				
REQ3				*		
REQ4			*			
REQ5					*	
REQ6			*		*	*
REQ7		*				
REQ8						*

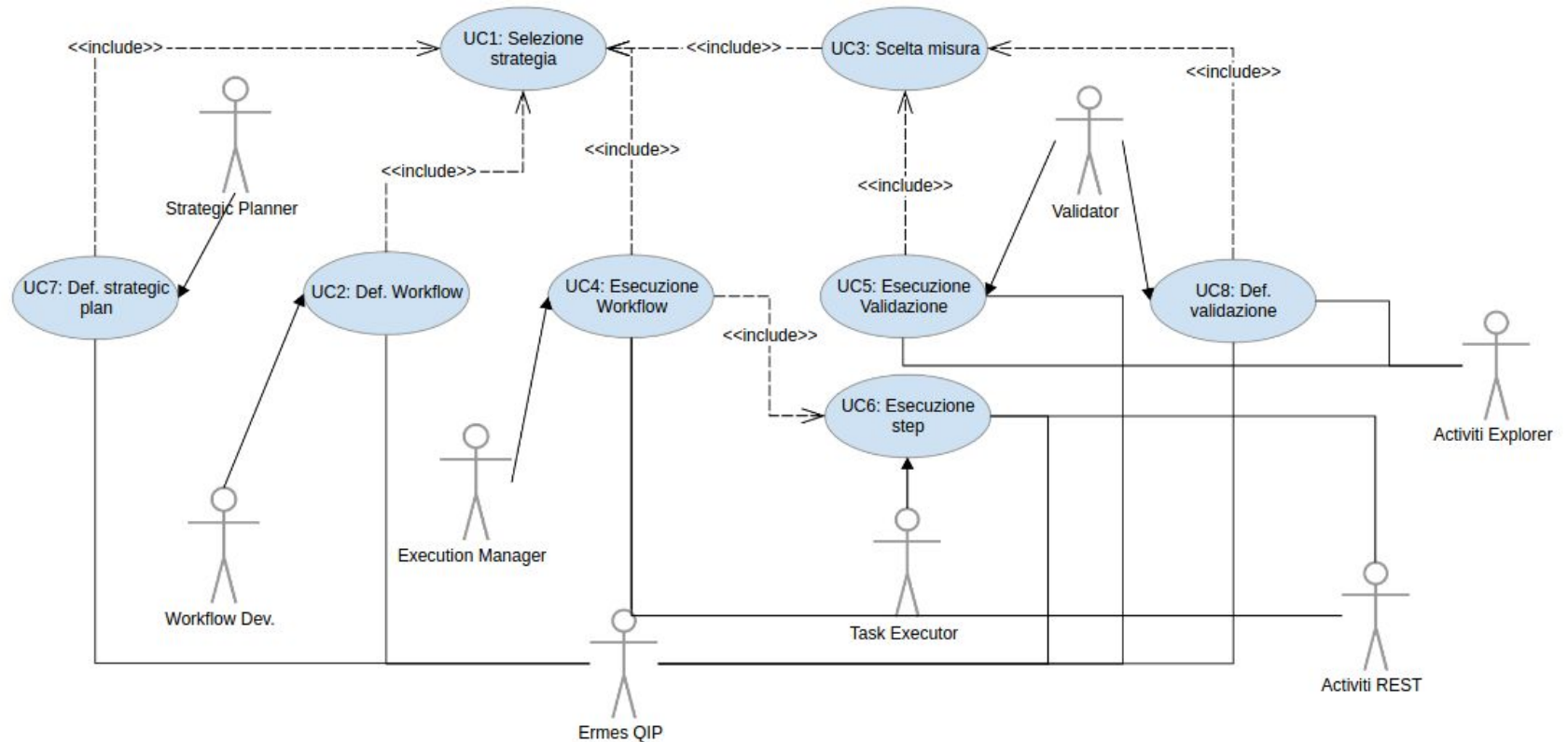


Use Cases

- UC1 - Selezione Strategia
- UC2 - Fase 3 - Definizione Workflow
- UC3 - Scelta misura
- UC4 - Fase 4 - Esecuzione Workflow
- UC5 - Fase 4 - Esecuzione Validazione
- UC6 - Fase 4 - Esecuzione Step
- UC7 - Fase 3 - Definizione Strategic Plan
- UC8 - Fase 3 - Definizione metodologie di validazione

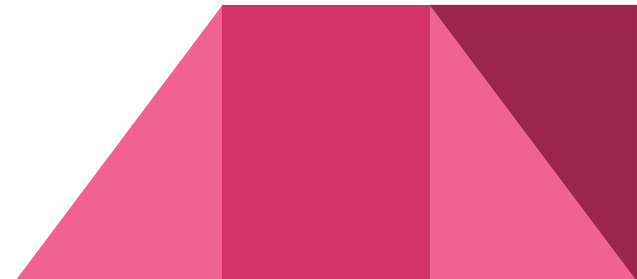


Use Case Diagram



Da Features a Use Cases

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
FEAT1	*						*	
FEAT2	*	*						
FEAT3	*			*		*		
FEAT4	*		*					*
FEAT5	*		*		*			
FEAT6	*	*	*	*	*	*	*	*



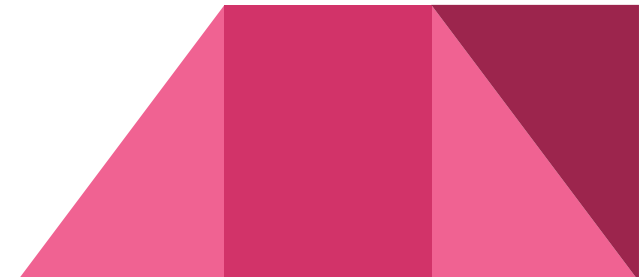
UC1 - Selezione Strategia (1)

- Attori coinvolti

- Strategic Planner
- Workflow Developer
- Execution Manager
- Validator
- Ermes QIP (non attivante)

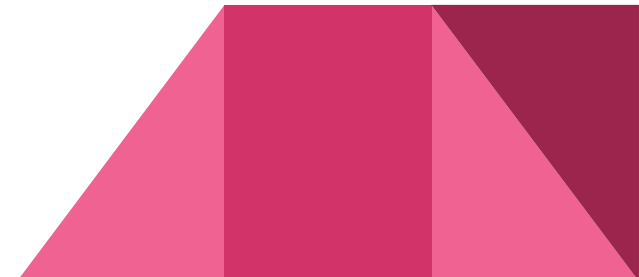
- Flusso base

- Ermes QIP fornisce la lista delle strategie
- L'utente sceglie la strategia
- Il sistema consente all'utente di selezionare un nuovo caso d'uso in base al suo ruolo.



UC1 - Selezione Strategia (2)

- Precondizioni
 - Strategie già definite
- Postcondizioni
 - Strategic Plans terminati



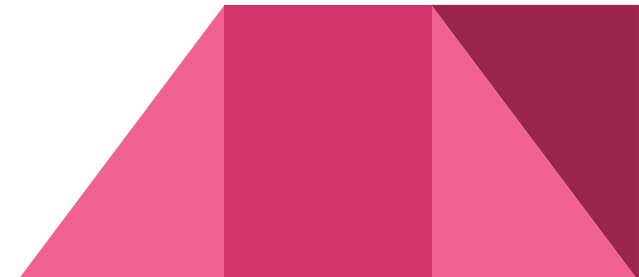
UC2 - Fase 3 - Definizione Workflow (1)

- Attori coinvolti

- Workflow Developer
- Activiti-Explorer

- Flusso base

- Il workflow developer crea o modifica il workflow.
- Il sistema memorizza il workflow.
- Se ci sono ancora workflow da definire, il sistema torna al punto 2.



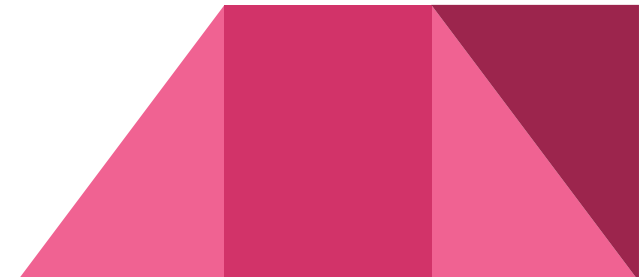
UC2 - Fase 3 - Definizione Workflow (2)

- Precondizioni

- Deve essere stato completato il caso d'uso relativo alla definizione dello strategic plan, e il workflow developer deve aver selezionato lo strategic plan per cui definire il workflow.

- Postcondizioni

- I workflow sono stati definiti per questa iterazione e si può quindi procedere alla definizione della validazione.



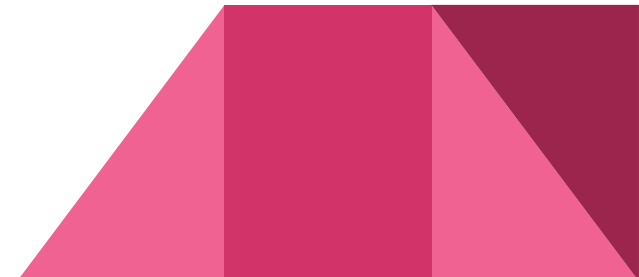
UC3 - Scelta Misura (1)

- Attori coinvolti

- Validator
- Ermes QIP

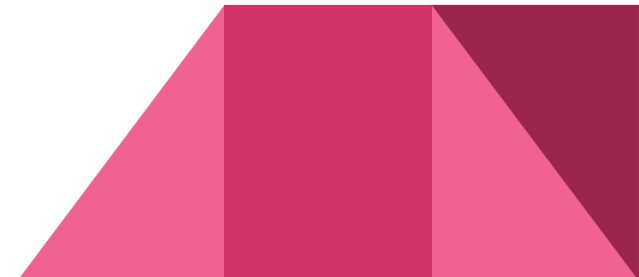
- Flusso base

- Il sistema mostra la lista delle misure definite all'interno di un workflow relativo ad una strategia.
- Il validator seleziona una misura.
- Il sistema consente all'utente di attivare un nuovo caso d'uso in base alla fase corrente.



UC3 - Scelta Misura (2)

- Precondizioni
 - Misure già definite
 - Selezione strategia
- Postcondizioni
 - Definizione della validazione della misura



UC4 - Fase 4 - Supervisione dell'esecuzione del Workflow

- Attori coinvolti
 - Execution Manager
 - Activiti-REST
 - Flusso base
 - Il sistema mostra una lista dei workflow attivi nell'iterazione corrente.
 - L'Execution Manager seleziona un workflow.
 - Il sistema visualizza lo stato di esecuzione del workflow.
 - Se l'Execution Manager vuole visualizzare lo stato di altri workflow, torna al passo 2.
 - Precondizioni
 - Fase 3 terminata
 - Postcondizioni
 - Esecuzione workflow terminata
- 

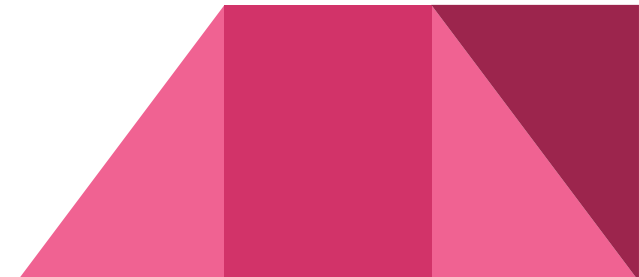
UC5 - Fase 4 - Esecuzione Validazione (1)

- Attori coinvolti

- Validator
- Ermes QIP

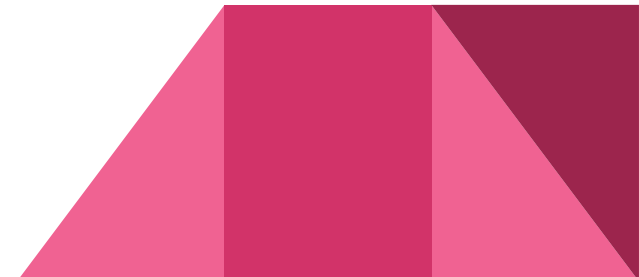
- Flusso base

- Il sistema visualizza una lista di operazioni di validazione, per una determinata misura, eseguite in maniera automatica.
- Il validator seleziona una operazione.
- Il sistema mostra le modifiche che sono state apportate dalla fase di validazione dei dati.
- Il validator revisiona le modifiche eseguite in modo automatico dalla componente di validazione, e le approva opportunamente.



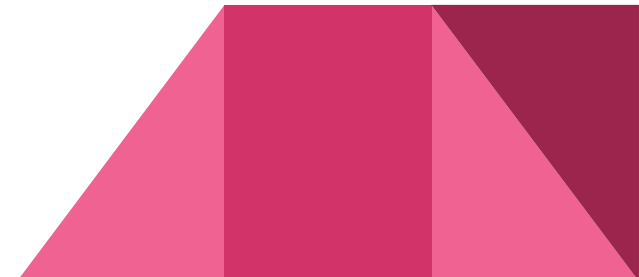
UC5 - Fase 4 - Esecuzione Validazione (2)

- Il validator revisiona le modifiche eseguite in modo automatico dalla componente di validazione, e le approva opportunamente.
- **Flusso alternativo**
 - All'ultimo passo del flusso base, se vi sono altre operazioni di validazione da revisionare, il Validator torna al primo.
- **Precondizioni**
 - Validazione dei dati automatica terminata
 - Selezione misura
- **Postcondizioni**
 - Validazioni approvate



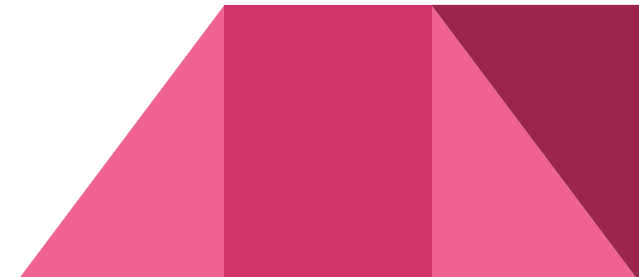
UC6 - Fase 4 - Esecuzione Step

- Attori coinvolti
 - Task executor
 - Attività REST
- Flusso base
 - Il sistema mostra una lista di task assegnati al task executor
 - Il Task Executor seleziona un task da svolgere.
 - Il sistema mostra il task da svolgere.
 - Il Task Executor riempie i campi richiesti e segna il task come completato
- Precondizioni
 - Fase 4 avviata
- Postcondizioni
 - Fase 4 terminata




UC7 - Fase 3 - Definizione Strategic Plan

- Attori coinvolti
 - Strategic Planner
 - Ermes QIP
- Flusso base
 - Il sistema mostra i possibili attributi definibili.
 - Lo strategic planner definisce gli attributi del piano strategico.
 - Il sistema memorizza il piano strategico.
 - Il sistema termina la fase di definizione dei piani strategici.
- Precondizioni
 - Strategia selezionata
- Postcondizioni
 - Strategic plans terminati



UC8 - Fase 3 - Definizione metodologie di validazione

- Attori coinvolti
 - Validator
 - Ermes QIP
 - Flusso base
 1. Il sistema mostra le operazioni di validazione presenti per quella misura, se ce ne sono.
 2. Il Validator aggiunge una nuova operazione di validazione.
 3. Il sistema mostra una serie di operazioni possibili e i relativi parametri da definire.
 4. Il Validator definisce i parametri.
 5. Il sistema associa l'operazione di validazione alla misura.
 6. Se il Validator non vuole fare altre modifiche alla validazione della misura, il caso d'uso termina. Altrimenti, torna al punto 1.
 - Precondizioni
 - Deve essere stata selezionata la misura per cui eseguire la validazione.
 - Postcondizioni
 - Devono esser state definite tutte le validazioni delle misure associate ad un workflow.
- 

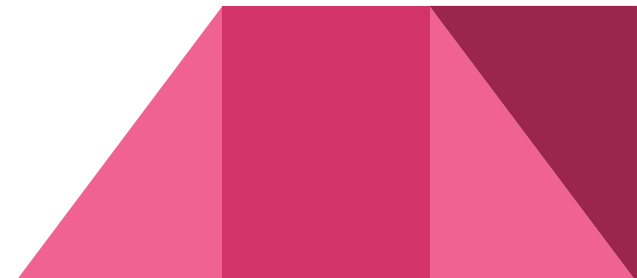
Da Casi d'Uso a Storie

- In Scrum le storie nascono dall'interazione diretta con il Product Owner, con grande enfasi sul dialogo tra le parti
- Nel nostro particolare caso le storie sono state create suddividendo i casi d'uso
- Se un caso d'uso ha più attori attivanti, viene generata una storia per ogni attore
- Le storie sono state ulteriormente scomposte per essere più piccole possibili

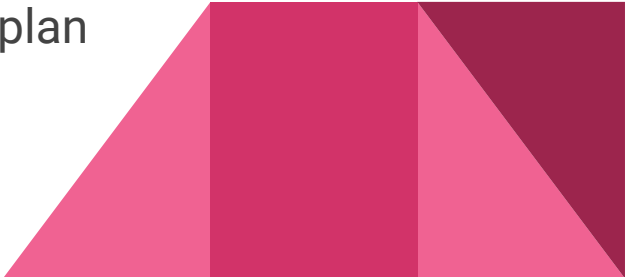


Da Casi d'Uso a Storie - Difficoltà riscontrate

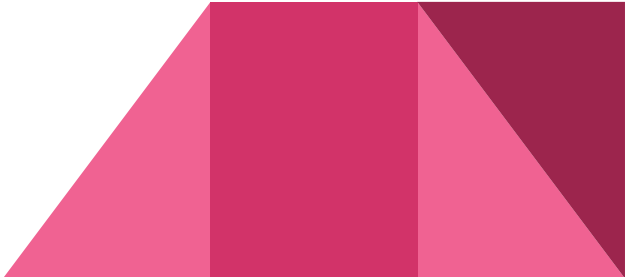
- La suddivisione in storie ha risentito dei cambi di direzione intrapresi nello sviluppo
- Sono stati generati molti task per riadattare storie già completate alla nuova architettura
- Inoltre probabilmente sarebbero state più opportune storie più piccole e meglio gestibili
 - Da tenere però in considerazione lo studio delle tecnologie
- Difficoltà nell'assegnare effort e priorità



User stories

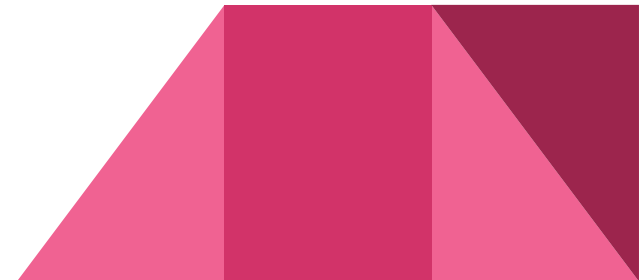
- Come strategic planner voglio visualizzare la lista delle strategie per poterle modificare in seguito
 - Come strategic planner voglio aggiungere degli attributi ad una strategia per poter creare il piano strategico ad essa associato
 - Come workflow developer voglio visualizzare una lista di strategic plans per poterne poi definire un workflow
 - Come workflow developer voglio creare o modificare il workflow relativo ad uno strategic plan per specificarne l'esecuzione
 - Come validator voglio visualizzare una lista di strategic plans per poterne definire la validazione
 - Come validator voglio visualizzare metodologie di validazione settate su una data misura, in modo tale da specificare lo strategic plan
- 

User stories (2)

- Come execution manager voglio visualizzare la lista degli strategic plans per poter supervisionare l'esecuzione del workflow associato
 - Come validator voglio selezionare una delle misure associate ad un workflow per poterne definire la validazione
 - Come execution manager voglio selezionare uno dei workflow attivi per poterne visualizzare lo stato di esecuzione
 - Come validator voglio aggiungere una metodologia di validazione ad una data misura, in modo tale da specificare lo strategic plan
 - Come validator voglio modificare una metodologia di validazione ad una data misura, in modo tale da specificare lo strategic plan
 - Come validator voglio eliminare una metodologia di validazione ad una data misura, in modo tale da specificare lo strategic plan
- 

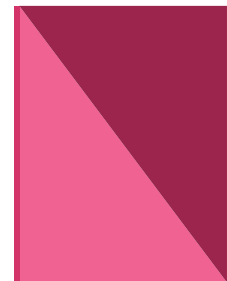
User stories (3)

- Come strategic planner voglio modificare degli attributi di una strategia per poter modificare il piano strategico ad essa associato
- Come task executor voglio eseguire un task a me assegnato per portare avanti l'esecuzione del workflow
- Come task executor voglio eseguire un task a me assegnato per terminare l'esecuzione del workflow



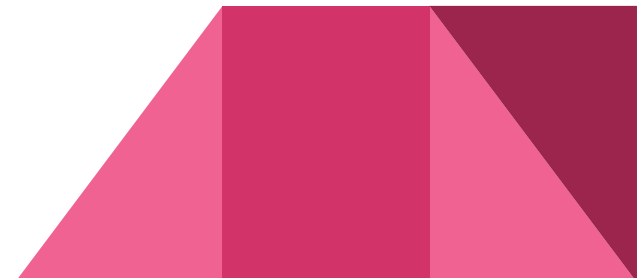
Divisione del lavoro - Daniele Capri

- Collaborazione nella definizione astrazioni chiave
- Pianificazione sprint
- Integrazione con Git
- Creazione e gestione dei modelli, servizi e chiamate rest relative agli strategic plan
- Gestione dei modelli, servizi e chiamate rest relative agli strategy
- Front-end di gestione dello strategic planner
- Interfaccia comunicazione 3.1 per la creazione di metaworkflow

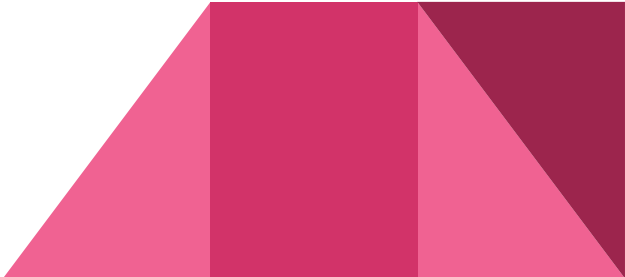


Divisione del lavoro - Daniele Capri

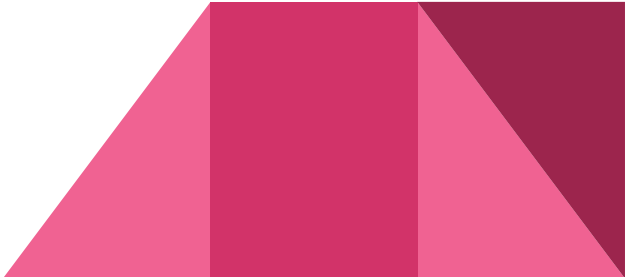
- Responsabile deploy
- Responsabile database
- Interazione con i gruppi delle fasi 0, 1, 2.2 e 5, 6
- Integrazione con il bus



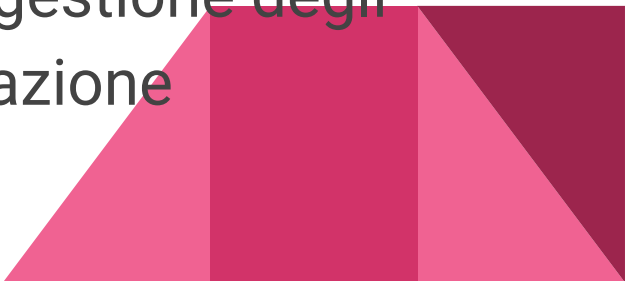
Divisione del lavoro - Giuseppe Chiapparò

- Definizione requisiti e features.
 - Stesura casi d'uso e storie.
 - Definizione astrazioni chiave.
 - Progetto e realizzazione **modifiche ad Activiti Explorer**.
 - Studio teorico e model della **validazione**.
 - Front-end della pianificazione della validazione.
 - Front-end dell'esecuzione della validazione.
 - Coordinamento della fase implementativa e di test sulla parte della validazione.
- 


Divisione del lavoro - Gabriele Belli

- Definizioni attori e stakeholders.
 - Revisione dei front end prima versione.
 - Implementazione da modello della validazione.
 - Implementazione servizi Rest per la Validazione.
 - Implementazione servizi Rest per la Measure Task.
 - Front end di Validazione.
 - Front end di Measure Task.
 - Implementazione servizio Rest per il nome dei task di misura attivi in Activiti.
- 

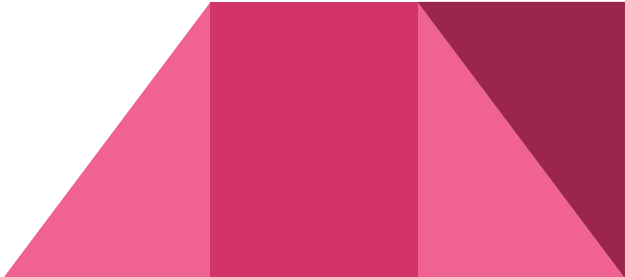
Divisione del lavoro - Fabio Alberto Coira

- Collaborazione nella stesura casi d'uso e storie
 - Collaborazione nella definizione astrazioni chiave
 - Pianificazione sprint
 - Studio di Activiti-REST e interazione da SpringBoot
 - Servizi REST per il dialogo con Activiti-REST
 - Servizi REST esposti al gruppo 3.1 e 4.1
 - Servizi REST per la Validazione
 - Prima versione Frontend e backend del Task Execution, Workflow Definition e Visualizzazione delle strategies
 - Struttura del progetto, manutenzione e gestione degli errori e delle eccezioni, e della configurazione
- 

Divisione del lavoro - Luca Della Gatta

- Collaborazione nella definizione astrazioni chiave
 - Front end creazione, visualizzazione, modifica Strategic Plan
 - Front end visualizzazione Strategie
 - Front end stato delle istanze dei processi attivi in Attività
 - Servizio rest per lo stato delle istanze dei processi attivi in Attività; creazione, modifica e visualizzazione dei Strategic Plan; visualizzazione strategie; visualizzazione strategie
 - Integrazione con Git, gestione dei conflitti
 - Integrazione front end con web app grafica
 - model del piano strategico e strategie
 - Documentazione servizi REST esposti per strategie e strategic plan
- 

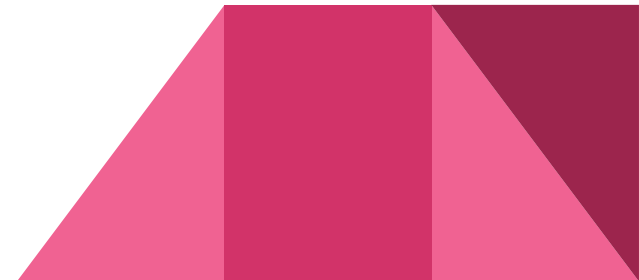
Divisione del lavoro - Federico Di Domenicantonio

- Definizione requisiti e features
 - Stesura casi d'uso e storie
 - Definizione astrazioni chiave
 - Pianificazione sprint
 - Integrazione con Git, gestione dei conflitti
 - Progetto e realizzazione modifiche ad **Activiti Explorer**
 - Studio teorico e model della **validazione**
 - Front end Task Execution
 - Front end Workflow Definition
 - Studio test case GQM⁺Strategies
- 

Key Abstraction

Piano strategico e strategie:

- Un piano strategico è un aggregazione di attributi che permettono di definire il piano strategico.
- Un piano strategico è legato ad una o più strategie.
- Ad un piano strategico sono associati 0 o più workflow.
- Una strategia può essere associata ad al più un workflow.

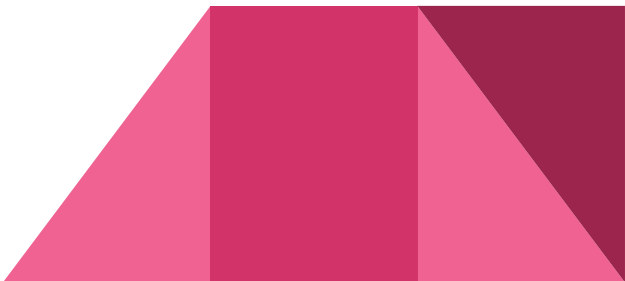


Key Abstraction

MeasuredWorkflow:

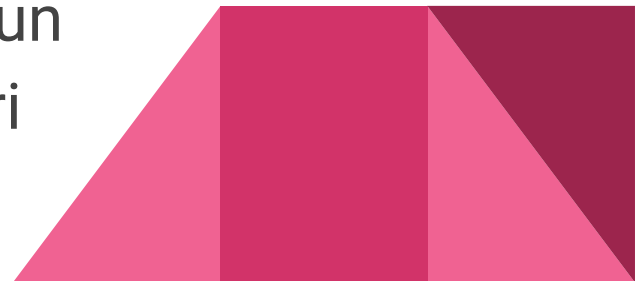
- Un measuredWorkflow estende Workflow.
- È un'aggregazione di metriche.
- Ad ogni metrica è associato un CollectedData.

ValidatedWorkflow:

- Un ValidatedWorkflow estende un MeasuredWorkflow.
 - È un'aggregazione di ValidatedMetric.
 - ValidatedMetric è l'aggregazione di una o più ValidationOp.
 - ValidationOp aggrega una o più Countermeasure.
 - Un ValidatedData è l'estensione di un CollectedData.
- 

Dettagli implementativi (1)

- Si è utilizzato SpringBoot esponendo degli endpoint attraverso dei RestController, uno per ogni “contesto” ritenuto necessario (es. attiviti, validationOp, etc)
- Implementazione di un certo numero di Services per offrire la logica di Business
- Customizzazione dell’ errore secondo un formato concordato con l’altro gruppo (attraverso implementazione di ErrorController)
- Intercettazione di eccezioni attraverso un ExceptionHahdler (introduzione di errori Custom)



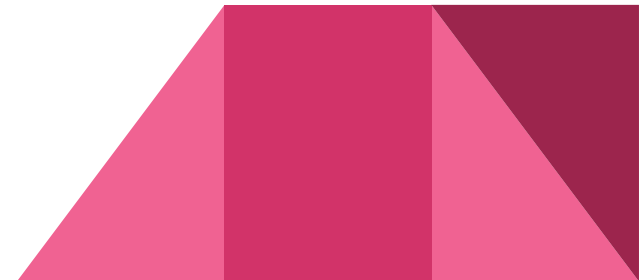
Dettagli implementativi (2)

- Utilizzo del Pattern DTO per il trasferimento dati
- Utilizzo dei Pojo e serializzazione/deserializzazione tramite Jackson
- Utilizzo di RestTemplate con la variante “.exchange” per il dialogo con Activiti-Rest (problema della Basic Authentication e della customizzazione dell’ header)
- Autoconfigurazione attraverso application.properties



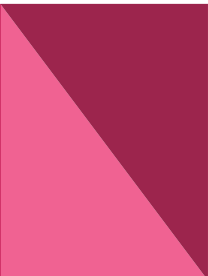
Conclusioni

- Occasione di studio di tool di workflow management (Activiti)
- Integrazione del codice e delle specifiche cross-team, con focus sulla comunicazione
- Adattamento a modifiche pesanti all'architettura dell'applicazione e riconfigurazione del team, nello spirito dell'approccio Agile
- Implementazione di un'applicazione RESTful
- Uso di database non relazionali
- Utilizzo del framework Spring



Sviluppi futuri

- Invio di un dato alla volta
- Integrità delle relazioni tra mongo ed Activity
- Implementazione delle strategie alternative
- Miglioramento dell'interazione tra il sistema e lo strategic planner
- Ottimizzazioni tipi di validazione
- Riuscire a risolvere in maniera adeguata il problema del Modeler
- Esportare completamente tutte le features di Explorer nel nostro sistema



Un caso reale di applicazione reale

Ora verrà mostrata una panoramica sull'esecuzione dell'applicazione, nel contesto del miglioramento della didattica all'interno dell'Università.

