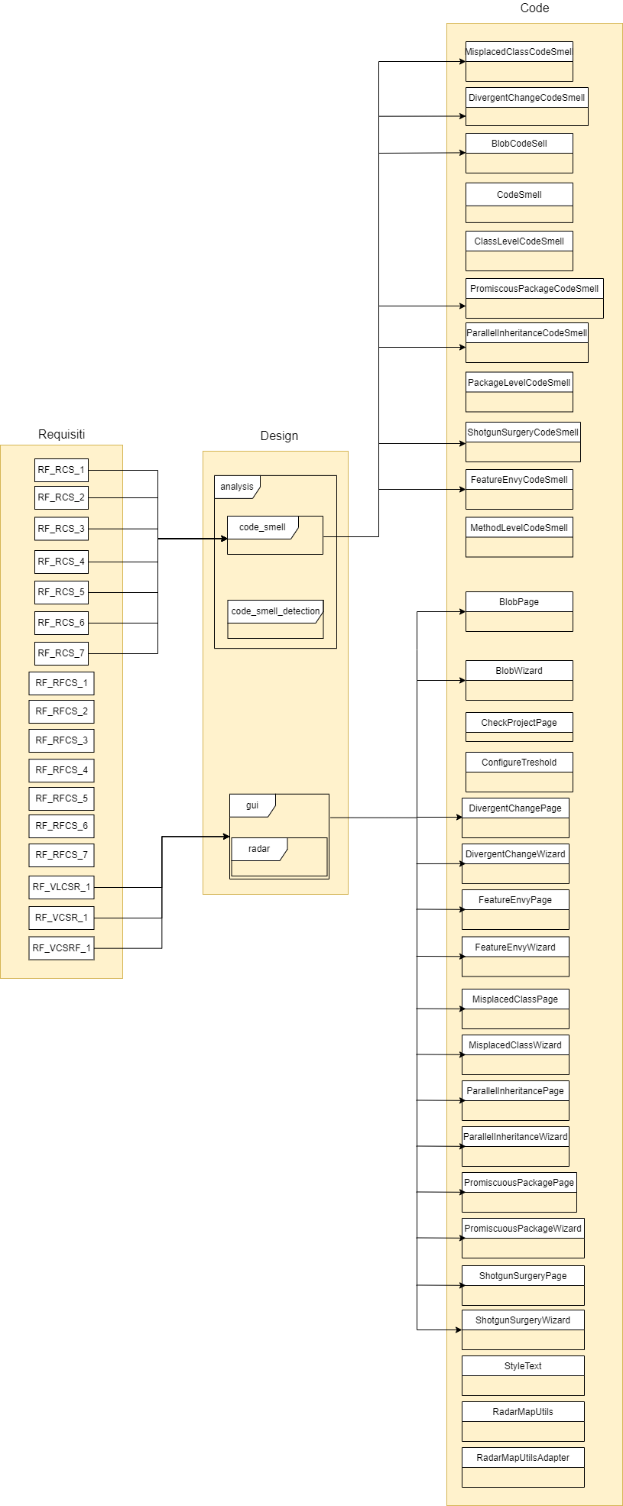
Impact Analysis

* **Identificazione Start impact set (SIS)**

L’impact analysis è iniziata col costruire un grafo della tracciabilità andando ad analizzare gli artefatti disponibili di cASpER. Il grafo risultante è il seguente (Per una migliore visualizzazione è disponibile nella cartella /Immagini/TraceabilityGraph):



Siamo partiti dai requisiti inserendo nell’omonima pila tutti i requisiti funzionali offerti da cASpER. Di tali requisiti nella change request vengono impattati solamente quelli relativi al rilevamento dei code smell e quelli relativi alla visualizzazione dei code smell presenti e alla visualizzazione dei refactoring.

Pertanto nella pila dedicata al design sono stai inseriti solamente i moduli che gestiscono tali requisiti, ovvero, Analysis e GUI. La tracciabilità orizzontale è stata mantenuta tra **RF\_VLCSR\_1**, **RF\_VCSR\_1**, **RF\_VCSRF\_1** e l’intero modulo GUI. Mentre per **RF\_RCS\_1**, **RF\_RCS\_2**, **RF\_RCS\_3**, **RF\_RCS\_4**, **RF\_RCS\_5**, **Rf\_RCS\_6**, **RF\_RCS\_7** la tracciabilità orizzontale è stata mantenuta solamente col sotto modulo code smell del modulo analysis in quanto il pattern visitor vuole essere applicato all’operazione di detection che viene eseguita nel modulo code smell, mentre il modulo code smell detection gestisce solamente le strategie di detection.

Dopo aver ottenuto la tracciabilità orizzontale tra la pila dei requisiti e quella del design siamo passati alla pila del codice, inserendo in essa tutte le classi del modulo code smell e GUI. Per la tracciabilità orizzontale tra la pila di design e codice sono stati escluse le classi Code Smell, ClassLevelCodeSmell, MethodLevelCodeSmell, PackageLevelCodeSmell, CheckProjectPage, ConfigureTreshold, StyleText, RadarMapUtils e RadarMapUtilsAdapter. Le prime tre vengono escluse in quanto non implementano il metodo per la detect degli smell ma viene implementato nelle classi che le estendono, quindi applicando il pattern visitor non saranno impattate. Tutte le altre vengono escluse in quanto non saranno impattate dal pattern Abstract Factory poiché esso lavora solamente sulle classi Wizard e Page.

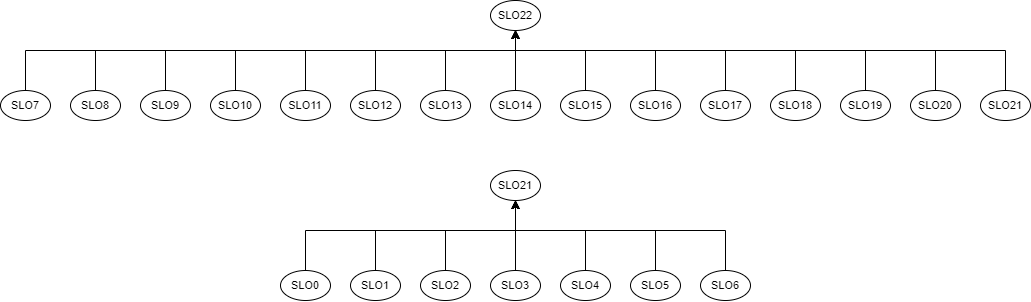
Dopo la pila del codice dovrebbe essere presente la pila dei test per capire quali test vengono impattati dalla CR, tuttavia in cASpER non è implementato nessun test per per la detection dei code smell ne per la GUI.

In conclusione lo start impact set (SIS) risulta essere il seguente, {**MisplacedClassCodeSmell, DivergentChangeCodeSmell, BlobCodeSmell, PromiscuousPackageCodeSmell, ParallelInheritanceCodeSmell, ShotgunSurgeryCodeSmell, FeatureEnvyCodeSmell, BlobPage, BlobWizard, DivergentChangePage, DivergentChangeWizard, FeatureEnvyPage, FeatureEnvyWizard, MisplacedClassPage, MisplacedClassWizard, ParallelInheritancePage, ParallelInheritancheWizard, PromiscuousPackagePage, PromiscuousPackageWizard, ShotgunSurgeryPage, ShotgunSurgeryWizard**}

* **Identificazione Candidate Impact Set (CIS)**

In tale fase integriamo il SIS con gli SLO, che possono cambiare a causa di modifiche agli elementi del SIS, analizzando gli impatti diretti e indiretti

|  |  |
| --- | --- |
| * SLO0 - MisplacedClassCodeSmell * SLO1 - DivergentChangeCodeSmell * SLO2 - BlobCodeSmell * SLO3 - PromiscuousPackageCodeSmell * SLO4 - ParallelInheritanceCoseSmell * SLO5 - ShotgunSurgeryCodeSmell * SLO6 - FutureEnvyCodeSmell * SLO7 - BlobPage * SLO8 - BlobWizard * SLO9 - DivergentChagePage * SLO10 - DivergentChangeWizard * SLO11 - FutureEnvyPage | * SLO12 - FutureEnvyWizard * SLO13 - MisplacedClassPage * SLO14 - MisplacedClassWizard * SLO15 - ParallelInheritancePage * SLO16 - ParallelInheritanceWizard * SLO17 - PromiscuousPackagePage * SLO18 - PromiscuousPackageWizard * SLO19 - ShotgunSurgeryPage * SLO20 - ShotgunSugeryWizard * SLO21 - PsiParser * SLO22 – CheckProjectPage |



Dopo aver costruito i vari grafi di connettività è stata costruita la matrice di tracciabilità. Essa è disponibile nel file excel Matrice Tracciabilità.

In conclusione il candidate impact set previsto per l implementazione di tale change request risulta essere il seguente, {**MisplacedClassCodeSmell, DivergentChangeCodeSmell, BlobCodeSmell, PromiscuousPackageCodeSmell, ParallelInheritanceCodeSmell, ShotgunSurgeryCodeSmell, FeatureEnvyCodeSmell, BlobPage, BlobWizard, DivergentChangePage, DivergentChangeWizard, FeatureEnvyPage, FeatureEnvyWizard, MisplacedClassPage, MisplacedClassWizard, ParallelInheritancePage, ParallelInheritancheWizard, PromiscuousPackagePage, PromiscuousPackageWizard, ShotgunSurgeryPage, ShotgunSurgeryWizard, PsiParser, CheckProjectPage**}

* **Actual Impact Set (AIS), Discovered Impact Set (DIS) e False Positive Impact Set (FPIS)**
* AIS

|  |  |
| --- | --- |
| ClassLevelCodeSmell  CodeSmell  MethodLevelCodeSmell  PackageLevelCodeSmell  MisplacedClassCodeSmell  DivergentChangeCodeSmell  BlobCodeSmell  PromiscuousPackageCodeSmell  ParallelInheritanceCoseSmell  ShotgunSurgeryCodeSmell  FutureEnvyCodeSmell  BlobPage  BlobWizard  DivergentChagePage  DivergentChangeWizard  FutureEnvyPage | FutureEnvyWizard  MisplacedClassPage  MisplacedClassWizard  ParallelInheritancePage  ParallelInheritanceWizard  PromiscuousPackagePage  PromiscuousPackageWizard  ShotgunSurgeryPage  ShotgunSugeryWizard  PsiParser  CheckProjectPage  ClassBean  MethodBean  Package Bean |

* DIS

|  |
| --- |
| ClassLevelCodeSmell  CodeSmell  MethodLevelCodeSmell  PackageLevelCodeSmell  ClassBean  MethodBean  Package Bean |

**Recall Impact Analysis:** |CIS ∩ AIS| /|AIS| = 23/30 = 0,76

**Precision Impact Analysis:** |CIS ∩ AIS| /|CIS| = 23/23 = 1