

# Verslag SWOP Groep 08: 28 maart

---

## 1 Aanpassingen op basis van feedback

Deze zijn allemaal doorgevoerd, mits de nodige refactoring. Modelexception is hernoemd naar ReportErrorToUserException.

## 2 Aanpassingen aan de lijst van use cases

We gebruiken geen reflection meer voor de lijst van mogelijke use cases voor een type User. Alle use cases zitten aparte klassen en erven van de abstracte klasse UseCase over.

## 3 Aanpassingen de search methods

We maken voor de zoekmethoden gebruik van het **strategy pattern**.

Bugreportservice bevat niet meer verschillende methodes om bugreports op te zoeken maar heeft een methode "search" dat een interface ISearch mee krijgt als argument.

Ipv. Bugreportservice :: public List<BugReport> searchOnTitle() ; Bugreportservice :: public List<BugReport> searchOnFiledUser() ; ... maken we nu gebruik van:

```
Bugreportservice :: public List<BugReport> search(ISearch search) { return search.apply(this); }
```

Waarbij er verschillende klassen zijn die ISearch implementeren (SearchOnTitle implements ISearch, SearchOnFiled implements ISearch, ...)

Er wordt dus steeds de juiste strategie meegegeven aan BugReportService en die geeft dan naargelang de strategie de lijst van BugReports.

Elke strategie wordt aangemaakt in de controller/usecase naargelang de keuze van de gebruiker. Deze aanpak zorgt niet voor meer koppeling omdat er al koppeling is vanuit de controller/usecases met de juiste bugreport zoekmethodes. Een strategie kan anderzijds niet uitgevoerd worden zonder de tussenkomst van BugReportService; ze zijn protected. Dit verhindert dat de strategie wordt toegepast vanuit de controller/usecases omdat dit de taak is van BugReportService. Dus: om cohesie hoog te houden, moet BugReportService deze strategie dan ook effectief toepassen en niet vanuit controller/usecases.

## 4 Design keuzes omtrent Mailbox, Notifications, Observer

Een mailbox behoort tot een Issuer. Mailbox is een Observer van een tag verandering in een specifieke bugreport of een comment verandering in een specifieke bugreport.

Dus moet mailbox zowel ObserverTag als ObserverComment implementeren die beiden overerven van de Observer interface. Jammer genoeg laat Java dit niet toe, dus hebben we dit opgelost met innerclasses.

ObserverTag, ObserverComment ... zijn de innerclasses van mailbox. Deze oplossing zorgt jammer genoeg wel voor een nogal hoge koppeling tussen mailbox en de verschillende observers.

Als er een nieuwe observer bijkomt, moet deze ook toegevoegd worden in de mailbox class. **Is deze koppeling aanvaardbaar ?** Indien niet, hoe kunnen we koppeling verlagen rekening houdend met Java Constraints ?

Een bugreport bevat comments, elke comment kan eventueel ook nog comments bevatten. Bij een toevoeging van een comment c2 aan comment c1 die uiteindelijk behoort bij een bepaalde bugreport, moet de corresponderende bugreport ook verwittigd worden. Dus lijkt het ons voor de hand liggend dat comment zowel een observer is van zijn children comments als een subject voor zijn comment observer of bugreport observer. **Mogen we deze assumptie effectief maken?**

Hetzelfde idee wordt ook gebruikt voor subsystem en project :: ze zijn beide zowel subject als observer. Bij een verandering van comment/tag worden de bovenstaande observers verwittigd door de hele hiërarchie terug te doorlopen.

Uiteindelijk wordt de notify ontvangen in observerTag/observerMail die dan respectievelijk een mail van het juiste type aanmaakt in de mailbox.

Dit zorgt wel voor meer koppeling omdat comments al een verwijzing hebben naar hun children comments, maar nu moet de children comments ook een verwijzing hebben naar hun observers (omwille van notifyObservers).

Dit geldt ook voor subsystem/bugreport en zorgt uiteindelijk op meerdere plekken een bidirectionele binding i.p.v. de oorspronkelijk unidirectionele binding. **Is deze verhoogde koppeling aanvaardbaar ?** Indien niet, hoe kunnen we de koppeling verlagen ? Is trouwens de Observer pattern hier de gepaste pattern, ondanks dat deze de koppeling nogal verhoogt ?

Dit heeft wel het voordeel dat bij toevoeging van nieuwe type observers in de mailbox, de hiërarchie eronder (project/subsystem/bugreport/comment) ongewijzigd blijven.

(zie bijlage voor high level code, Class diagram en sequence diagram)

## 5 Andere functionaliteiten

Over de design keuzes van onze andere delen zullen we volgende week maandag (04/04) rapporteren.