Egon a quelques problèmes comme méthode pour repérer les classes divines. Elle est polluée par d’autre classes qui ne sont pas divine mais auront des caractéristiques semblables.

Le premier type de class qu’on remarque est le « main » . Dans notre cas JFreeChart.java est appelé le plus souvent, soit 650 fois. Les contrôleurs auraient probablement un problème semblable. On s’attend à ce qu’il soit plus couplé que d’autres méthodes. Il serait probablement une bonne idée de tout de même les analyser, mais ils sortiront probablement toujours dans le haut de la courbe de distribution. Ces méthodes ne font qu’appeler les autres paquets pour communiquer le flow des étapes. Lors de l’analyse, il faut porter une attention particulière pour être sûr que si on modifie le code pour limiter le nombre de ligne et le coupable qu’on englobe par du code qui n’a pas lieu d’être. (Non testable et /ou complétement dépendant du flow du main ou du contrôleur)

Le deuxième type de classe qui sortira lors de l’analyse sont les classes utiles. Une classe utile est créer pour faciliter l’écriture du code et englobé des problèmes qui reviennent souvent. Par exemple, des classes qui travailles des types de données spéciaux vont souvent revenir. Dans notre cas, nous avons TextUtils, SerialUtils et DatasetUtils qui sont sortient.

Des classes utiles peuvent parfois être grand en quantité de lignes, car elles doivent être bien documentés, pour faciliter l’utilisation et certaines d’entre elles nécessitent plus méthodes pour englober spécifications. Il faudrait donc peut-être changer le calcul du nombre de ligne pour calcul de nombre de ligne de code.

Ceci nous apporte à un autre point important de Egon, les commentaires. Si un projet est réalisé par plusieurs développeurs qui ne commentent pas de la même façon ( et que certain ne commentent pas ). La courbe de répartition du nombre de lignes sera influencée. Certain code avec beaucoup de ligne de code pourrait ne par sortir dû à des package écris de façon plus concise mais avec plus de commentaire.

En conclusion, Egon a besoin de quelques améliorations et ne sera probablement jamais parfais. Mais cette méthode peut toujours être utile pour repérer des classes divines si le développeur l’utilisant le fait d’un œil critique.