C# Eindopdracht **(Rhyzo)** reflectieverslag

In dit reflectieverslag bespreek ik mijn ervaringen die ik heb opgedaan en inzichten tijdens het ontwikkelen van Rhyzo. Ik wil hier reflecteren op mijn zwakke punten en op de sterke punten van het project en om te kijken naar verbeteringen.

# Projectbeschrijving

**Rhyzo** is een muziek-stream console app, waarbij de user vanuit de console de opties heeft om nummers af te spelen, afspeellijsten te creëren, albums toevoegen aan afspeellijsten etc.

# Projectplanning

De planning van het project verliep tot op het laatste moment. Ik ben erin geslaagd om de meeste Definition of Done (DoD) criteria van de user stories te realiseren. Echter, tijdens het ontwikkelen heb ik vaak mijn gedachten veranderd over de structuur van mijn code. Gelukkig kon ik hierbij gebruik maken van ChatGPT en adviezen van vrienden met betrekking tot de OOP structuur. Het heroverwegen en aanpassen van de code-structuur heeft veel tijd gekost, wat invloed heeft gehad op het realiseren van de DoD’s.

# Codekwaliteit

De codekwaliteit vind ik mij betreft over het algemeen goed. Ik heb gebruik gemaakt van het **Singleton design pattern.** Het Singleton pattern zorgt ervoor dat er slechts een instantie van een class wordt gecreëerd gedurende runtime van de applicatie. De code is (volgens mij) leesbaar en gestructureerd, en ik heb commentaar toegevoegd bij de functies die belang hebben bij de werking van het project. Ik had meer aandacht kunnen besteden aan de UI gedeelte (wat ik ook van plan was) maar niet heb kunnen redden i.v.m. tijd en slordig planning.

# Wat heb ik geleerd?

In de loop van de periode dat we aan dit project moesten werken heb ik heel veel geleerd:

1. Hoe het Singleton design pattern in elkaar zit
2. Hoe dictionary’s werken en hoe array structuren in het algemeen in elkaar zitten
3. Meer logischer gaan denken door te debuggen doormiddel van f11 (line by line)
4. Een beter beeld gekregen van hoe SCRUM werkt en hoe je ermee moet werken **(een leerpunt voor mijn volgende project).**
5. Hoe je efficiënt kun refereren naar classes

# Verbetervoorstel

Over het algemeen ben ik tevreden met het eindresultaat van het project. Er zijn wel wat verbeterpunten die ik in de toekomstige projecten zou willen aanpakken:

1. Meer aandacht voor het plannen i.p.v. het coderen
2. Meer aandacht geven aan een goede UML schema maken
3. Meer aandacht geven aan uitgebreider testen van scenario’s
4. Niet aan teveel user stories werken in een keer
5. Deadline neerzetten voor backlogs.