Voor en nadelen

Nieuwe tool versus uitbreiden

# Doel van dit document

Het doel van dit document is om stakeholders van de MES configuratie tool ervan bewust te maken wat op huidig tijdstip waar de meeste voordelen uit de halen zijn voor de applicatie, maar ook om te laten zien waar op dit moment de knel punten zitten.

Het gaat hierbij vooral om de toekomstbestendigheid van de applicatie en dus onderhoud- en uitbreidbaarheid.

# Voor en nadelen Uitbreiden

|  |  |
| --- | --- |
| **Voordeel** | **Nadeel** |
| De applicatie is er al en hij werkt soms | Geen MVVM |
| Sommige dingen werken al | Fouten opsporen gaat zeer moeizaam |
|  | Geen goede WPF /.xaml implementatie |
|  | Geen goede architectuur |
|  | Geen goede consistentie |
|  | Spelfouten |
|  | Slechte GUI implementatie |
|  | Niet tot weinig getest. Mocht dit nog gedaan moeten worden dan is dit bijna niet te doen |
|  | Hele slechte performance |
|  | GUI elementen in de businesslaag (waardoor de GUI implementatie op verschillende locaties staat) |
|  | WPF is gebruikt maar de applicatie had net zogoed in forms gebouwt kunnen zijn aangezien technieken uit WPF nauwelijks gebruikt zijn |

# Voor en nadelen Nieuwe opzet

|  |  |
| --- | --- |
| **Voordeel** | **Nadeel** |
| WPF MVVM vanaf het begin (GUI en Business zijn apart aanpasbaar) | Veel functionaliteiten zal opnieuw over na gedacht moeten worden. |
| GUI zal meteen goed geïntegreerd worden | Nieuwe technieken worden geïmplementeerd. Deze zullen tijd kosten om te leren. |
| Beter toekomstbestendig |  |
| Betere architectuur opzet |  |
| Veel betere DAL implementatie |  |
| Er worden daad werkelijk klassen met relaties daartussen gebruikt (dus overzichtelijker en makkelijker in gebruik) |  |
| Proof of concept met basis voor de applicatie is al gemaakt |  |
| Meteen goede SAD en overdrachtsdocumentatie |  |
| Unit of work patroon (geen fatale database fouten) |  |

# Persoonlijke voorkeur

Het is bijna onbegonnen werk om de applicatie in zijn huidige staat uit te breiden of te refactoren. Dit zal naar mijn inschatting meer kosten dan mijn stage periode en daar zit deze stagiair niet op te wachten.

Verder verdiep ik mezelf liever in de nieuwe technieken van WPF i.p.v. de oude forms technieken toe te passen zoals huidig het geval is.

Doormiddel van de technieken in combinatie met een nieuwe kans voor de configuratie applicatie is mijn inschatting dat er veel minder tijd vereist is om een nieuwe opzet te maken met een betere architectuur dan de huidige applicatie om te bouwen.

Dit komt vooral door het feit dat niemand weet hoe de huidige configuratie applicatie nou precies in elkaar steekt (inclusief ikzelf).

De applicatie KAN omgebouwd worden. Dat is inderdaad een optie, maar hiervoor zullen gehele stukken en misschien bijna de gehele opnieuw gebouwd moeten worden. Op zo’n moment kun je je afvragen of het niet veel slimmer is om opnieuw te beginnen met een duidelijk een GOED voorbereid plan na dat er ONDERZOEK is verricht over implementatie technieken die waarschijnlijk enorm handig zijn om te gebruiken in de applicatie waardoor fatale fouten voorkomen kunnen worden en stukjes code gemakkelijker terug gevonden kunnen worden

# Technieken waaruit voordelen gehaald kunnen worden

|  |  |
| --- | --- |
| **Techniek** | **Beschrijving** |
| MVVM (Model-View-ViewModel) | Zorgt ervoor dat GUI layer en Business layer gescheiden blijven |
| Entity framework | Zorgt ervoor dat database statements niet meer deels in de database geïnjecteerd kunnen worden. (alles faalt of alles gaat goed) |
| .xaml | Huidige applicatie staat veel GIU elementen in de business laag. Deze moeten allemaal in de .xaml komen te staan zodat alle GUI logica op een plek staat |
| Behaviors | WPF .xaml implementatie waarmee zelfgemaakte UI controls gebouwd kunnen worden. Hiermee wordt er meer logica uit de business laag naar de GUI laag gezet |
| Klassen met relaties | Er zijn wel klassen op het moment, maar bij de meeste is niet te zien ui9t het klassen diagram wat het nut er van is en hoe ze in elkaar zitten, met relaties er tussen wordt het nut en de bewerkingen van de individuele klasse een stuk duidelijker |
| Data & command binding | Hier wordt op het moment nauwelijks gebruik van gemaakt. Door data en command binding is het zeer eenvoudig om functionaliteiten en de GUI opmaak aan te passen of te verplaatsen. |