# Code Review

### Code Style Analyse

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Code Richtlinien von HSR Video Wall wurden eingehalten | X |  |
| Der Code wirkt durch seine Anordnung & Verschachtelung übersichtlich | X |  |
| Die HSR Video Wall Headers sind in allen nicht generierten Sourcen vorhanden | X |  |
| Die XML-Kommentar Kompilation wurde in den Projekten aktiviert Problem nur im DiagnosticApp, das nicht zum Code Review gehört | X |  |
| Alle Public / Protected Members sind ausreichend Dokumentiert  Problem nur im DiagnosticApp, das nicht zum Code Review gehört | X |  |
| Die auskommentierten Programm-Stücke sind ausreichend erklärt | X |  |
| Die Projekte enthalten keine toten Programm-Klassen | X |  |
| Die fehlenden Programmstücke sind mittels TODO-Kommentar beschrieben | X |  |
| Der Code übersetzt ohne Compiler Warnings, die nicht dokumentiert sind und sich nicht in den Test Projekten befinden | X |  |
| Es gibt keine Bad Smells im Code Diverse Kleine sind noch vorhanden. | X |  |

### Exception Handling

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Der Code enthält keine abgefangenen und ignorierten Ausnahmen | X |  |
| Fehler in asynchronen Prozessen werden mittels Event weitergeleiltet | X |  |
| Das Logging erfasst alle Fehler aus allen Funktionalitätsschichten | X |  |
| Das Before/After Pattern wird, wo möglich, mittels using() { } angewendet | X |  |
| IDisposable.Dispose() Methoden werden in jedem Fall aufgerufen | X |  |
| Alle Elemente, die aufgeräumt werden müssen, implementieren das Interface IDisposable  Wie besprochen; Noch nicht komplett implementiert |  | X |

### Flow Control

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Es existieren keine Schleifen ohne Abbruchkriterien | X |  |
| Es existieren keine nicht dokumentierte, tote Programmstücke (z.B. if(false) / while (false) / …) | X |  |
| Rekursive Calls haben immer eine Verankerung und Abbruchbedingung | X |  |

### Naming

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Namen der Klassen / Variabeln sind selbstbeschreibend | X |  |
| Verwirrende oder falsche Namen sind nicht vorhanden | X |  |
| Interface-Klassen beginnen immer I (z.B. IDisposable) | X |  |
| Klassen / Properties / Methoden werden mit *PascalCasing* geschrieben | X |  |
| Lokale Variablen / Argumente werden mit *camelCasing* geschrieben | X |  |
| Konstanten werden in *PascalCasing* geschrieben | X |  |
| Felder in Klassen werden mit *\_camelCasing* geschrieben | X |  |
| Der Code enthält keine Magic Numbers | X |  |

### Tools

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Warnings von Resharper 4.XXX werden, wo sinnvoll, behoben  Die „disabled Warnings“ vom Reshaper stören den Lesefluss im Programmcode erheblich. Zusätzlich besteht die Gefahr, dass bei neuen Anpassungen gerade weitere Unschönheiten durch die Bestehenden „disabling“-Anweisung ausgeblendet werden.  Daher: Lieber nicht disablen und im Source (z.B. als Remark / …) beschreiben, warum dies so sein muss als zukünftige Implementationen erschweren.  Ansonsten ist dies ok. | X |  |
| Die Errors von FxCop werden wo sinnvoll behoben | X |  |