Iteration I – EPIC 1.1

**Sprint-Planning-**

Prioritäten/Aufwandsschätzung nach Folge(1,3,5,8,10, BIG\_INT)

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition der Stories** | **Aufwandsschätzung** | **Tasks** | **Verantwortliche** |
| Epic 1.1 Implementierung Abspielen Ton | 8 |  |  |
|  |  | Methode playSound | Team I |
|  |  | Black-Box Test | Team II |

Sprint-Ziel

Abspielen des Sounds.

DoD

Erfolgreiches Testen und Abnahme des Product-Owners.

Daily-Scrum

Team I: Probleme bei Konstrukt, Aufbau der Klassen. PlaySound Methode noch nicht implementiert. CallMethod Parameter nicht korrekt.

Team II: Ausführbare Klasse. Objekt inkl. Zugriff und playDemo Aufruf erstellt. Probleme: wrapperID als Übergabeparameter nicht korrekt.

Sprint-Review

Team I: playSound erfolgreich implementiert. Lösungsansatz: mit callMethod. Grundverständnis für Firmware Implementierung bekommen, Firmware Zugriff mit play Methode und in playSound umgesetzt.

Team II: Ausführbare Demo Klasse erfolgreich implementiert. Test durchgeführt. InternetButton spielt sound.

Product-Owner: Abnahme, bei Test anwesend. Aufgabe erfüllt.

Sprint-Retroperspektive

Verbesserungs-Maßnahmen: Struktur/Aufbau 🡪 Firmware Methoden-Wissen und Zugriffe.

Iteration II – EPIC 1.2 -1.3

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition der Stories** | **Aufwandsschätzung** | **Tasks** | **Verantwortliche** |
| Epic 1.2 Led ON/OFF (Switch) | 5 |  |  |
|  |  | Methode setzen/abschalten | Team I |
|  |  | Test ohne Button | Team I |
|  |  | Button-Zugriff: Zeigen Led Zugriff | Team I |
| Epic 1.3 Zähler setzen/zurücksetzen | 5 |  |  |
|  |  | Zähler setzen/zurücksetzen | Team II |
|  |  | Applikation Zähler auslesen /zurücksetzen | Team II |

Sprint-Ziel: I

Implementierung der Funktionen LED on/off sowie die Möglichkeit einen Button-Zähler, zur Dokumentation über Bedienung, zu implementieren.

DoD

Erfolgreiches Testen und Abnahme des Product-Owners.

DailySrum

Team I: Methode setLed implementiert. Mit CallMethod die Firmware Methode Led übergeben sowie RGB Werte gesetzt. Demo Klasse ebenfalls erfolgreich implementiert

Team II Methode ResetButtonCounter implementiert, Button 1-4 zählt nach Ausführung der Test Applikation korrekt.

Sprint-Review

Team I und Team II erfolgreich Epic 1.2 und 1.3 vorgeführt.

Sprint-Retroperspektive

Zusammenarbeit der Teams besser strukturieren, Abstimmung Commit und Pull Operationen. 🡪 Zeitvorgabe oder manuelles Kommando von Teams. Problem in Iteration II: Test ausgeführt ohne Pull – kein aktueller Stand, daher Annahme Fehler der Implementierung.

Iteration III – EPIC 2.1 – 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition der Stories** | **Aufwandsschätzung** | **Tasks** | **Verantwortliche** |
| Epic 2.1 Klickzähler | 5 |  |  |
|  |  | Klicks zählen & Position zählen übergeben | Team I |
|  |  | Test Applikation | Team I |
| Epic 2.2 Klasse Rotanteile | 3 |  |  |
|  |  | Rotanteile verändern /steigern | Team II |
|  |  | Test Applikation | Team II |

Sprint-Ziel: I

Implementierung eines Klickzählers sowie einer eigenen Klasse wo Rotanteile aufsteigend verändert/gesteigert werden.

DoD

Erfolgreiches Testen und Abnahme des Product-Owners.

Daily-Srum

Team I: Implementierung vollständig. Ausgeführt, getestet. Commit Probleme auf einem PC Programmierer I. Epic 2.1 befindet sich in ButtonCounter Demo 🡪 ausführbare Klasse

Team II: Probleme mit Ausführung/ DemoApp. Neue Klasse ColorShade lässt keinen Zugriff in ausführender Klasse auf InternetButton Objekt. Klasse ColorShade fertig implementiert – wenn Button gedrückt wird, wird Rotwert um 10RGB Werte erhöht. Demo Klasse nicht erfolgreich implementiert.

Sprint-Review

Epics 2.1 und 2.2 erfolgreich implementiert, Product Owner hat an Vorführung teilgeneommen und erfolgreich abgenommen.

Sprint-Retroperspektive

Zusammenarbeit funktioniert bereits sehr gut 🡪 konnte in den letzten Iterationen verbessert werden. Teams-Einteilung funktioniert sehr gut. Übersicht über Struktur und Aufbau der Klassen wird vehement klarer für alle Programmierer.