

Message Logger

Projektmanagement-Plan

1. Projektorganisation	2
1.1. Organisationsplan, Rollen & Zuständigkeiten	2
1.2. Projektstrukturplan	2
2. Projektführung	3
2.1. Rahmenplan	3
2.2. Projektkontrolle	3
2.3. Risikomanagement.....	3
2.4. Projektabschluss	4
3. Projektunterstützung	5
3.1. Tools für Entwicklung, Test & Abnahme	5
Testplan	6
3.2. Testdesign & Abläufe	6
3.3. Testfälle	6
4. Anhänge.....	7
4.1. Paketdiagramm	7
4.2. Sprint1-Statistik	7
4.3. Sprint2-Statistik	7

Versionen:

Rev.	Datum	Autor	Bemerkungen	Status
0.1	25.09.2017	Melvin Werthmüller	1. Entwurf	Fertig
0.2	29.09.2017	Christopher Christensen	2. Rollen	Fertig
0.3	09.10.2017	Melvin Werthmüller	Projektunterstützung	Fertig
0.4	10.10.2017	Lukas Arnold, Melvin Werthmüller, Christopher Christensen	Rahmenplan	Fertig
0.5	16.10.2017	Valentin Bürgler	Projektkontrolle	Fertig
0.6	17.10.2017	Christopher Christensen, Melvin Werthmüller, Lukas Arnold	Testplan	Fertig
0.7	02.11.2017	Valentin Bürgler	Ergänzungen zu allen Kapiteln	Fertig
0.8	04.11.2017	Valentin Bürgler	Überarbeitung	Fertig
0.9	05.11.2017	Valentin Bürgler	Bereit für Zwischenabgabe	Fertig

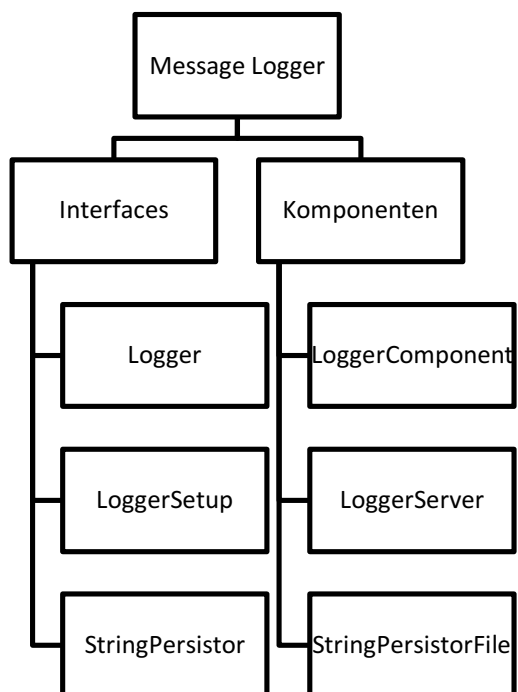
1. Projektorganisation

1.1. Organisationsplan, Rollen & Zuständigkeiten

Name	Aufgabe
Christopher Christensen	ProductOwner, Terminplanung
Valentin Bürgler	Scrum-Master, Dokumentation
Lukas Arnold	Interfacer, Code-Master
Melvin Werthmüller	Reporting, Projektleiter

1.2. Projektstrukturplan

Der Projektstrukturplan ist objektorientiert gegliedert in die folgenden Arbeitspakete:



2. Projektführung

2.1. Rahmenplan

Meilenstein	Beschreibung	Zeitpunkt
1	Organisation der Gruppe ist definiert (SoDa-Rollen); erste Risikoliste Produktbacklog und Sprintplanung für Sprint 1 liegen vor und sind im PMP dokumentiert.	SW 03
2	Dokumentationsplan. Liste der Konfigurations-Items. Spezifikation der drei Elemente für das Systemtesting einschliesslich der Definition des Vorgehens liegt vor. Entwicklung Sprint 1 abgeschlossen. Code wird in GitLab verwaltet und laufend integriert. Erste (geforderte) Unit-Tests laufen erfolgreich. Sprint 2 geplant (SprintBacklog).	SW 05
3	Demonstration / Präsentation (Zwischenabgabe Mo. 6./Di. 7. Nov. 2017) Sprint 2 abgeschlossen. Architektur ist festgelegt und exemplarisch dokumentiert. Release 1 ist lauffähig und kann demonstriert werden. Sprint 3 ist geplant (SprintBacklog). Vorgängig Abgabe der Dokumentation & Projektcontrolling (elektronisch) => So. 5. Nov. 2017, 18:00 ILIAS Briefkasten, Peer Review ist organisiert (personell und zeitlich).	SW 08
4	Sprint 4 abgeschlossen. Nachgeführte Softwarespezifikation liegt vor und ist reviewed. Alle Komponenten sind lauffähig und können demonstriert werden. Die Interoperabilität der Logger-Komponente kann demonstriert werden. Demonstration / Präsentation (Schlussabgabe Di. 12. Dez. 2017) Vorgängig Abgabe der Dokumentation (elektronisch + Papier). => Mo. 11. Dez. 2017, 18:00 ILIAS Briefkasten.	SW 13

2.2. Projektkontrolle

Der Soll/Ist Vergleich wird mittels Zeiterfassung der Work-Items ermittelt. Damit man erkennen kann, ob das Projekt planungsgemäss fortschreitet, verwendet man folgende Tools:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

2.3. Risikomanagement

Folgende Tabelle listet die für uns relevanten Risiken auf. Die Eintrittswahrscheinlichkeit wird mit der Skala von 1 (unwahrscheinlich) bis 10 (sehr wahrscheinlich) eingestuft. Das Schadensausmass wird von 1 (kein Schaden) bis 10 (erheblicher Schaden) eingeschätzt.

Nr.	Risikobeschreibung	Eintrittswahrscheinlichkeit	Schadensausmass	Summe
1	Datenverlust	2	8	16
2	Personenausfall	5	5	25
3	Änderungen der Requirements	10	5	50
4	Ausfall des Git-Servers	2	2	4
5	Ausfall von Ilias	2	1	2

Zusätzliche Bemerkung bei einer Gesamtsumme von über 20:

Nr.	Bemerkung
2	Ein Ausfall einer Arbeitskraft ist leider möglich. Die anfallende Arbeit muss dann auf die verbleibenden Personen aufgeteilt werden.
3	Dieses Projekt ist ein Schulprojekt und die Änderung der Requirments wurde angedeutet. Dadurch entsteht ein Mehraufwand, welchen wir irgendwie bewältigen müssen.

2.4. Projektabschluss

Der Projektabschluss ist am 11.12.2017 um 18:00 Uhr.

3. Projektunterstützung

3.1. Tools für Entwicklung, Test & Abnahme

Für die Entwicklung, inkl. dem Testing, wird IntelliJ in Kombination mit GitLab verwendet.

Um eine flexible Kommunikation zu ermöglichen, existiert ein WhatsApp Gruppen-Chat. Die meisten Klassen werden mit JUnit-Tests überprüft, wo sinnvoll (siehe Abbildung).

```
public class StringPersistorTest {  
  
    private static Logger LOG = LogManager.getLogger(StringPersistorTest.class);  
  
    @Test  
    public void setFile() throws Exception {  
        // Arrange  
        StringPersistor stringPersistor = new StringPersistor();  
        File file = new File("TestFile.txt");  
  
        // Act  
        stringPersistor.setFile(file);  
  
        // Assert  
        assertEquals("TestFile.txt", stringPersistor.getFile().getPath());  
  
        // Revert  
        LOG.info(file.delete());  
    }  
  
    @Test  
    public void save() throws Exception {  
        // Arrange
```

Testplan

3.2. Testdesign & Abläufe

Um die Funktionalität des Systems zu überprüfen werden die 4 folgenden Tests manuell durchgeführt. sZur Überprüfung der einzelnen Komponenten werden JUnit-Tests verwendet.

3.3. Testfälle

Case 1: Good Case

- Server starten
- Game starten
- Mehre Aktionen vornehmen
- Log-Einträge auf dem Server überprüfen

Case 2: Internetverbindung zwischendurch unterbrochen

- Server starten
- Game starten
- Mehrere Aktionen vornehmen
- Internetverbindung unterbrechen
- Mehrere Aktionen vornehmen
- Internetverbindung wiederherstellen
- Log-Einträge auf dem Server überprüfen (Aktionen während Internetunterbrechung)

Case 3: Sehr langsame Internetverbindung (Timeouts)

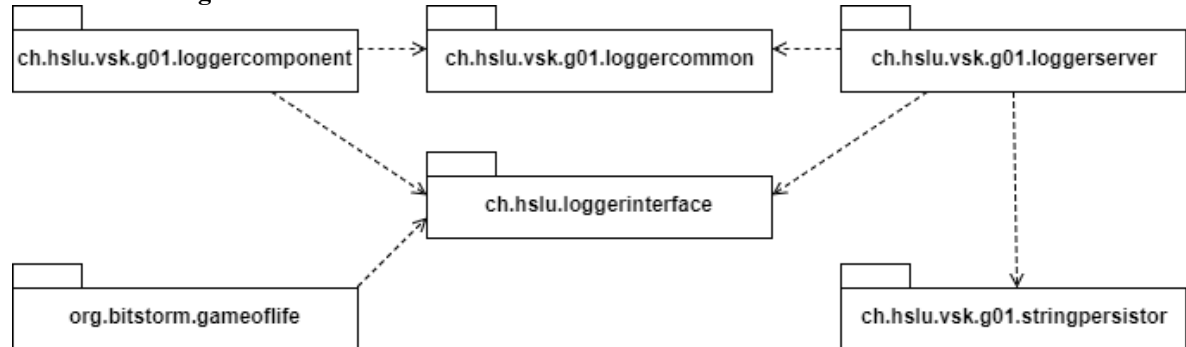
- Server starten
- Game starten
- Mehrere Aktionen vornehmen
- Internetverbindung verlangsamen
- Mehrere Aktionen vornehmen
- Log-Einträge auf dem Server überprüfen

Case 4: Good, Ersetzen der Logger Komponenten, Nochmals Testen

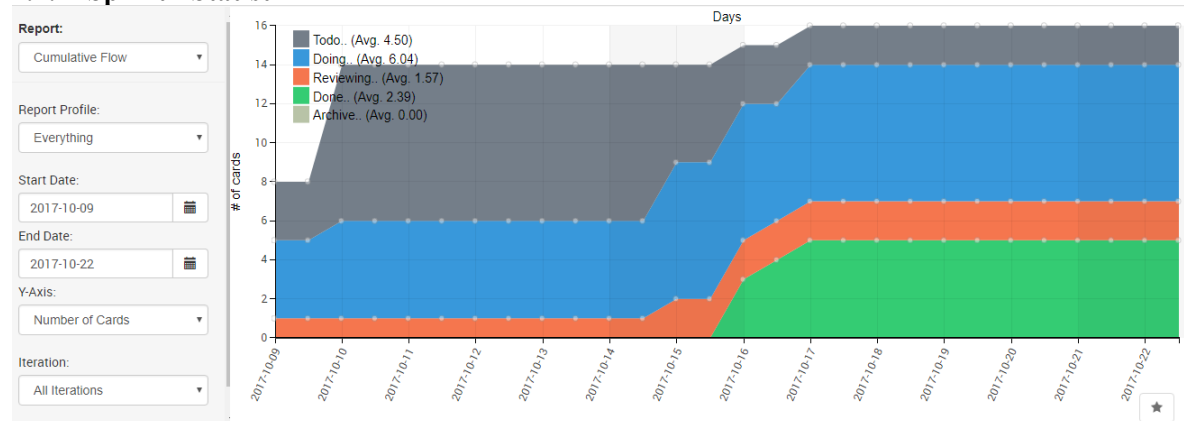
- Server starten
- Game starten
- Mehrere Aktionen vornehmen
- Log-Einträge auf dem Server überprüfen
- Game beenden
- Logger-Komponente austauschen
- Game starten
- Mehrere Aktionen vornehmen
- Log-Einträge auf dem Server überprüfen

4. Anhänge

4.1. Paketdiagramm



4.2. Sprint1-Statistik



4.3. Sprint2-Statistik

