Entwicklung einer Klassenbibliothek zur Erzeugung autokorrelierter Zufallszahlen

Projektplan

Abteilung Informatik

Hochschule für Technik Rapperswil

|  |
| --- |
| Herbstsemester 2017 |

Autor(en): Anthony Delay

Philipp Bütikofer

Betreuer: Prof. Dr. Andreas Rinkel

Lukas Kretschmann

# Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| <Datum> | <Nr> | <Art der Änderung> | <Vorname Name> |
|  |  |  |  |
|  |  |  |  |

# Inhalt

[Änderungsgeschichte 2](#_Toc494200124)

[Inhalt 3](#_Toc494200125)

[1. Einführung 4](#_Toc494200126)

[1.1 Zweck 4](#_Toc494200127)

[1.2 Gültigkeitsbereich 4](#_Toc494200128)

[1.3 Referenzen 4](#_Toc494200129)

[2. Projekt Übersicht 5](#_Toc494200130)

[2.1 Zweck und Ziel 5](#_Toc494200131)

[2.2 Annahmen und Einschränkungen 5](#_Toc494200132)

[3. Management Abläufe 6](#_Toc494200133)

[3.1 Kostenvoranschlag 6](#_Toc494200134)

[3.2 Zeitliche Planung 6](#_Toc494200135)

[3.2.1 Phasen 6](#_Toc494200136)

[3.2.2 Meilensteine 7](#_Toc494200137)

[3.3 Besprechungen 8](#_Toc494200138)

[4. Arbeitspakete 9](#_Toc494200139)

[5. Infrastruktur 11](#_Toc494200140)

[6. Qualitätsmassnahmen 12](#_Toc494200141)

[6.1 Dokumentation 12](#_Toc494200142)

[6.2 Projektmanagement 12](#_Toc494200143)

[6.3 Entwicklung 12](#_Toc494200144)

[6.3.1 Vorgehen 12](#_Toc494200145)

[6.3.2 Unit Testing 12](#_Toc494200146)

[6.3.3 Code Reviews 12](#_Toc494200147)

[7. Simulation 12](#_Toc494200148)

# Einführung

## Zweck

Dieses Dokument regelt den Ablauf und Aufbau des gesamten Projektes. Es soll einen Leitfaden für alle am Projekt beteiligten Personen darstellen.

## Gültigkeitsbereich

Dieses Dokument ist im Rahmen der Studienarbeit 2017 «Entwicklung einer Klassenbibliothek zur Erzeugung autokorrelierter Zufallszahlen gültig.

## Referenzen

|  |  |
| --- | --- |
| **Bezeichnung** | **Link** |
| Zeiterfassung |  |
| Sitzungsprotokoll |  |
| ToDo-Liste |  |

# Projekt Übersicht

Im Rahmen dieser Studienarbeit soll eine Klassenbibliothek zur Erzeugung von autokorrelierter Zufallszahlen entwickelt werden und anschliessend in die Simulationsumgebung Simio integriert werden.

## Zweck und Ziel

In der Simulation von Systemen werden Zufallszahlen zur Beschreibung der einzelnen Arbeitsschritte benötigt. Diese Zufallszahlen werden so erzeugt, dass sie keine Abhängigkeiten bzw. Autokorrelationen auftreten. Nun hat sich jedoch gezeigt, dass ebendiese Autokorrelationen in der Praxis häufig auftreten. Dadurch können die simulierten und realen Ergebnisse stark voneinander abweichen. Ziel dieses Projektes ist es, eine Klassenbibliothek für Simio zu entwickeln, welche den Grad der Autokorrelation einstellen lässt.

# Management Abläufe

## Kostenvoranschlag

<Wie viel Zeit steht zur Verfügung? In welcher Zeitspanne läuft das Projekt? Wird das Projekt früher beendet, dafür wöchentlich mehr gearbeitet?>

Zeitaufwand für SA analysieren

## Zeitliche Planung



### Phasen

|  |  |
| --- | --- |
| **Bezeichnung** | **Kickoff** |
| **Beschreibung** | * Vertraut machen mit dem Themengebiet * Projektorganisation * Recherche |
| **Zeitraum** | 18.09.2017 – 27.09.2017 |
| **Subphasen** | - |
| **Ziel** | * Projektmanagement ist operativ   + Git   + Zube   + Zeiterfassung   + Dokumentationsvorlagen   + Projektplan erstellt * Grobübersicht von ARTA erlangt |

|  |  |
| --- | --- |
| **Bezeichnung** | **Recherche** |
| **Beschreibung** | * Weitere Quellen zum Themenbereich ausfindig machen * Simio-Refresher |
| **Startdatum** | 27.09.2017 |
| **Enddatum** | 11.10.2017 |
| **Subphasen** | Analyse ARTA/JARTA  Analyse Integration Simio |
| **Ziel** | * Vertiefte Wissen über ARTA erlangen * JARTA: Code-Analyse * SIMIO-Schnittstellen kennen |

|  |  |
| --- | --- |
| **Bezeichnung** | **Implementation Klassenbibliothek** |
| **Beschreibung** | * Überführen des ARTA-Prozesses von Java nach C# * Testing des neu erzeugten Codes * Integration der Klassenbibliothek in Simio |
| **Startdatum** | 12.10.2017 |
| **Enddatum** | 26.10.2017 |
| **Subphasen** | Implementation & Testing der Bibliothek  Integration der Bibliothek in Simio |
| **Ziel** | * Klassenbibliothek des ARTA-Prozesses in C# * Schnittstelle zu SIMIO realisiert |

|  |  |
| --- | --- |
| **Bezeichnung** | **Simulation** |
| **Beschreibung** | * Simulationsumgebung erzeugen * ARTA-# in verschiedenen Simulationen testen bzw. vergleichen |
| **Startdatum** | 26.10.2017 |
| **Enddatum** | 22.11.2017 |
| **Subphasen** | Simulationsumgebung einrichten  Simulationen durchführen |
| **Ziel** | * Simulieren/Testen der Klassenbibliothek * Erzeugung von auszuwertenden Daten |

|  |  |
| --- | --- |
| **Bezeichnung** | **Auswertung und Dokumentation** |
| **Beschreibung** | * Erzeugte Daten auswerten * Überprüfen der erwarteten Resultate * Resultate und Ergebnisse dokumentieren |
| **Startdatum** | 23.11.2017 |
| **Enddatum** | 13.12.2017 |
| **Subphasen** | Simulationsauswertung  Dokumentation der Resultate |
| **Ziel** | * Dokumentation der erlangten Erkenntnisse/Resultate |

|  |  |
| --- | --- |
| **Bezeichnung** | **Präsentation und Abgabe** |
| **Beschreibung** | * Abschlussarbeiten * Falls nötig; Reservezeit * Binden und Drucken der Arbeit |
| **Startdatum** | 14.12.2017 |
| **Enddatum** | 22.12.2017 |
| **Subphasen** | - |
| **Ziel** | * Abgabe der Studienarbeit |

### Meilensteine

|  |  |
| --- | --- |
| **Bezeichnung** | **Setup komplett** |
| **Beschreibung** | Projektplan, Projektmanagement und administrative Vorbereitungen vollständig abgeschlossen |
| **Termin** | 11.10.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Abschluss Recherchen Arbeiten** |
| **Beschreibung** | Der ARTA-Prozess ist klar und verstanden, die Integrationsfragen für eigene Klassenbibliotheken in Simio sind geklärt, Code Analyse von JARTA vollständig abgeschlossen |
| **Termin** | 11.10.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Implementation der Klassenbibliothek** |
| **Beschreibung** | Die Implementation der Klassenbibliothek ist vollständig abgeschlossen, ein Code-Review ist durchgeführt |
| **Termin** | 26.10.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Simulationsumgebung vorbereitet** |
| **Beschreibung** | Eine sinnvolle Simulationsumgebung ist innerhalb von Simio erzeugt |
| **Termin** | 08.11.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Simulationen durchgeführt** |
| **Beschreibung** | Simulationen basierend auf der erzeugten Klassenbibliothek und Simulationsumgebung durchgeführt |
| **Termin** | 22.11.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Auswertung der Resultate** |
| **Beschreibung** | Die Resultate der vorgängigen Simulation sind ausgewertet |
| **Termin** | 13.12.2017 |

## Besprechungen

Sämtliche, reguläre, Besprechungen für das gesamte Zeitfenster der SA sind bereits fixiert worden. Sie finden jeweils am Mittwoch statt, wobei zwischen zwei Arten von Sitzungen unterschieden wird: Zwischenstandabklärung und offizielle Sitzung. Die Zwischenstandsitzungen finden jeweils im SA-Raum statt.

Für jede Besprechung wir ein Sitzungsprotokoll geschrieben (siehe Referenzen). Zusätzlich wird eine projektweite ToDo-Liste geführt, welche jedem Sitzungsprotokoll angehängt wird.

|  |  |  |
| --- | --- | --- |
| **Datum** | **Zeit** | **Räumlichkeit** |
| 27.09.2017 | 13:30 | 8.225 |
| 04.10.2017 | 13:30 | 1.262 |
| 11.10.2017 | 13:30 | 8.225 |
| 18.10.2017 | 13:30 | 1.262 |
| 25.10.2017 | 13:30 | 8.225 |
| 30.10.2017 | 13:30 | 1.262 |
| 8.11.2017 | 13:30 | 8.225 |
| 15.11.2017 | 13:30 | 1.262 |
| 22.11.2017 | 13:30 | 8.225 |
| 29.11.2017 | 13:30 | 1.262 |
| 06.12.2017 | 13:30 | 8.225 |
| 13.12.2017 | 13:30 | 1.262 |
| 20.12.2017 | 13:30 | 8.225 |

# Arbeitspakete

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Bezeichnung** | **Startdatum** | **Enddatum** |
| **Kickoff** | Zube einrichten |  |  |
| Git einrichten |  |  |
| Projektplan erstellen |  |  |
| Dokumentationsvorlagen erstellen |  |  |
| Zeiterfassungsliste in Excel erstellen |  |  |
| Einlesen in Papaer: JARTA |  |  |
| Einlesen in Paper: ARTA |  |  |
| Feinplanung nächste Phase |  |  |
| **Recherche** | Analyse ARTA-Prozess |  |  |
| Rechercheunterstützung HSR-Bibliothek |  |  |
| Analyse JARTA |  |  |
| Code-Analyse JARTA |  |  |
| Refresher Simio |  |  |
| Feinplanung nächste Phase |  |  |
| **Implementation** | UML-Klassendiagramm |  |  |
| Unit-Tests |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Feinplanung nächste Phase |  |  |
| **Simulation** |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Feinplanung nächste Phase |  |  |
| **Auswertung** |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Feinplanung nächste Phase |  |  |
| **Abgabe** |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Infrastruktur

Im Rahmen der SA sind uns pro Person je ein Arbeitsplatz mit Computer zur Verfügung gestellt worden.

|  |  |
| --- | --- |
| **Raum** | **Arbeitsplätze** |
| 1.262 | 14/15 |

Folgende Tabelle zeigt alle während der SA eingesetzten Tools.

|  |  |
| --- | --- |
| **Bezeichnung** | **Produkt** |
| Entwicklungsumgebung | Visual Studio 2017  Eclipse Java Mars |
| Versionskontrolle | Github |
| Simulationsumgebung | Simio |
| Projektmanagement | Zube.io |

# Qualitätsmassnahmen

## Dokumentation

Das gesamte Projekt (Dokumentation, Code, sämtliche Files) ist auf Github abgelegt.

Link zum Git-Repository: <https://github.com/ntdelay/Semesterarbeit-HS-2017-2018>

## Projektmanagement

Für die Abwicklung des Projektes wird das Tool Zube genutzt. Darin werden die einzelnen Sprints und deren Arbeitspakete erfasst, welche wiederrum den entsprechenden Personen zugewiesen werden kann.

Link zum Zube-Projekt: [www.zube.io/SAI17](https://zube.io/sai17/entwicklung-einer-klassenbibliothek-zur-erzeugung-autokorrelierter-zufallszahlen/w/operativer-workspace/kanban)

Die Zeiterfassung erfolgt über eine separate Excelliste (siehe Referenzen), da Zube diese Funktionalität nicht mit sich bringt.

## Entwicklung

Der Source-Code befindet sich ebenfalls im Git-Repository des Projektes.

Link zum Git-Repository: <https://github.com/ntdelay/Semesterarbeit-HS-2017-2018>

### Unit Testing

<Wo werden welche Unit Tests geschrieben um die Qualität sicherzustellen? Wie wird die Testabdeckung sichergestellt (z.B. durch EclEmma)?>

Der Code wird durch Unit-Tests abgedeckt

Vorgegeben UNIT Tests, decken den ARTA Prozess ab

### Code Reviews

<Werden Code Reviews gemacht und wie werden diese gemacht?>

Ja eh, mit lukas

# Simulation

Ein grosser Teil der Arbeit umfasst die Simulation der Klassenbibliothek. Einerseits wird dadurch die Integration in Simio getestet, andererseits die Richtigkeit der erzeugten Werte.

Dient zur Verifikation der Umsetzung von ARTA

Dient zur Validation der Integration von ARTA-# innerhalb von Simio