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 Kretschmar

# Inhalt

[Inhalt 2](#_Toc500922578)

[1. Einführung 3](#_Toc500922579)

[1.1 Zweck 3](#_Toc500922580)

[1.2 Gültigkeitsbereich 3](#_Toc500922581)

[1.3 Referenzen 3](#_Toc500922582)

[2. Projekt Übersicht 3](#_Toc500922583)

[2.1 Zweck und Ziel 3](#_Toc500922584)

[3. Management Abläufe 4](#_Toc500922585)

[3.1 Kostenvoranschlag 4](#_Toc500922586)

[3.2 Zeitliche Planung 5](#_Toc500922587)

[3.2.1 Phasen 6](#_Toc500922588)

[3.2.2 Meilensteine 8](#_Toc500922589)

[3.3 Besprechungen 9](#_Toc500922590)

[4. Arbeitspakete 10](#_Toc500922591)

[5. Infrastruktur 12](#_Toc500922592)

[6. Qualitätsmassnahmen 12](#_Toc500922593)

[6.1 Dokumentation 12](#_Toc500922594)

[6.2 Projektmanagement 12](#_Toc500922595)

[6.3 Entwicklung 12](#_Toc500922596)

[6.3.1 Code Reviews & Unit Testing 12](#_Toc500922597)

[7. Simulation 13](#_Toc500922598)

# 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 | [https://github.com/ntdelay/Semesterarbeit-HS-2017-2018](https://github.com/ntdelay/Semesterarbeit-HS-2017-2018.git) |
| Sitzungsprotokoll | [https://github.com/ntdelay/Semesterarbeit-HS-2017-2018](https://github.com/ntdelay/Semesterarbeit-HS-2017-2018.git) |
| ToDo-Liste | [https://github.com/ntdelay/Semesterarbeit-HS-2017-2018](https://github.com/ntdelay/Semesterarbeit-HS-2017-2018.git) |

# 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.

Zusätzlich zu den wissenschaftlichen Zielen, möchten wir uns in folgenden Punkten selbst weiterbilden.

* Einarbeiten in eine neue Problemdomäne: ARTA
* Vertiefung der Kenntnisse: Simio, C#
* Erzeugen eines wissenschaftlichen Papers
* Vertiefung im Bereich Projektmanagement
* Erarbeitung einer Simulationsumgebung zur Verifikation eines komplexen Prozesses

# Management Abläufe

## Kostenvoranschlag

|  |  |
| --- | --- |
| **Zeitraum** | 14 Wochen |
| **Geplante Arbeitstage** | Montag: 8 h |
|  | Mittwoch: 4 h |
|  | Samstag: 4-8 h |
| **Total geplante Arbeitszeit:** | 14 Wochen \* 16 h/Woche |
|  | = 224 h / Person |

## Zeitliche Planung



### Phasen

|  |  |
| --- | --- |
| **Bezeichnung** | **Kickoff** |
| **Beschreibung** | * Vertraut machen mit dem Themengebiet * Projektorganisation * Recherche |
| **Zeitraum** | 18.09.2017 – 27.09.2017 |
| **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 |
| **Ziel** | * Vertiefte Wissen über ARTA erlangen * JARTA: Code-Analyse * SIMIO-Schnittstellen kennen |

|  |  |
| --- | --- |
| **Bezeichnung** | **Konzeption 1** |
| **Beschreibung** | * Überführen des ARTA-Prozesses von Java nach C# * Theorie: Autokorrelation * Aufbau der Dokumentation |
| **Startdatum** | 12.10.2017 |
| **Enddatum** | 25.10.2017 |
| **Ziel** | * Klassenbibliothek des ARTA-Prozesses in C# * Theoretische Aspekte zu Autokorrelation dokumentiert |

|  |  |
| --- | --- |
| **Bezeichnung** | **Konzeption 2** |
| **Beschreibung** | * Integration der Klassenbibliothek in Simio * Statistische Tests |
| **Startdatum** | 26.10.2017 |
| **Enddatum** | 08.11.2017 |
| **Ziel** | * Statistische Tests umgesetzt * Schnittstelle zu SIMIO realisiert |

|  |  |
| --- | --- |
| **Bezeichnung** | **Implementation Klassenbibliothek** |
| **Beschreibung** | * Überführen des ARTA-Prozesses von Java nach C# * Testing des neu erzeugten Codes * Integration der Klassenbibliothek in Simio * Realisierung der Konzeptionsphasen 1 & 2 |
| **Startdatum** | 9.11.2017 |
| **Enddatum** | 22.11.2017 |
| **Ziel** | * Klassenbibliothek des ARTA-Prozesses in C# * Schnittstelle zu SIMIO realisiert |

|  |  |
| --- | --- |
| **Bezeichnung** | **Testing** |
| **Beschreibung** | * Statistische Tests der Klassenbibliothek |
| **Startdatum** | 23.11.2017 |
| **Enddatum** | 06.12.2017 |
| **Ziel** | * Unit-Test abgedeckt * Statistische Tests realisiert |

|  |  |
| --- | --- |
| **Bezeichnung** | **Simulation und Auswertung** |
| **Beschreibung** | * Simulationsumgebung erzeugen * ARTA-# in verschiedenen Simulationen testen bzw. vergleichen |
| **Startdatum** | 07.12.2017 |
| **Enddatum** | 20.12.2017 |
| **Ziel** | * Simulieren/Testen der Klassenbibliothek * Erzeugung von auszuwertenden Daten |

|  |  |
| --- | --- |
| **Bezeichnung** | **Abgabe** |
| **Beschreibung** | * Abschlussarbeiten * Falls nötig; Reservezeit * Binden und Drucken der Arbeit |
| **Startdatum** | 21.12.2017 |
| **Enddatum** | 22.12.2017 |
| **Ziel** | * Abgabe der Studienarbeit |

### Meilensteine

|  |  |
| --- | --- |
| **Bezeichnung** | **Abschluss Recherchenarbeiten** |
| **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** | **Konzeption der Klassenbibliothek & Autokorrelation** |
| **Beschreibung** | Die Umsetzung der Klassenbibliothek ist klar definiert. Weiter sind theoretische Aspekte zu den Themen Autokorrelation und deren Auswirkung dokumentiert. |
| **Termin** | 25.10.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Konzeption Testing & Integration** |
| **Beschreibung** | Einbindung der Klassenbibliothek in Simio ist klar. Statistische Tests für den Code sind vorhanden. |
| **Termin** | 08.11.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Implementation** |
| **Beschreibung** | Die Implementation der Klassenbibliothek ist vollständig abgeschlossen, ein Code-Review ist durchgeführt |
| **Termin** | 22.11.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Testing komplett** |
| **Beschreibung** | Unit-Testing komplett. Statistische Tests ergeben erwartete Resultate. |
| **Termin** | 08.12.2017 |

|  |  |
| --- | --- |
| **Bezeichnung** | **Simulation & Auswertung** |
| **Beschreibung** | Simulationen basierend auf der erzeugten Klassenbibliothek und Simulationsumgebung durchgeführt  Die Resultate der vorgängigen Simulation sind ausgewertet |
| **Termin** | 20.12.2017 |

## 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

Noch provisorisch, im Laufe des Projekts werden Pakete noch spezifischer unterteilt.

|  |  |
| --- | --- |
| **Phase** | **Bezeichnung** |
| **Kickoff** | Zube einrichten |
| Git einrichten |
| Projektplan erstellen |
| Rechercheunterstützung HSR-Bibliothek |
| Dokumentationsvorlagen erstellen |
| Zeiterfassungsliste in Excel erstellen |
| Einlesen in Paper: JARTA |
| Einlesen in Paper: ARTA |
| Feinplanung nächste Phase |
| **Recherche** | Rechercheunterstützung HSR-Bibliothek |
| Analyse ARTA-Prozess |
| Dokumentation:  -ARTA, Zusammenfassung |
| Analyse JARTA |
| Code-Analyse JARTA |
| Dokumentation:  -JARTA, Zusammenfassung  -Code, Spezifikation |
| Refresher Simio Grundlagen |
| Recherche Simio: Einbinden einer Libary |
| Dokumentation:  -Planung: Umsetzung JARTA in c#  -Planung: Umsetzung ARTA # in Simio |
| Erfahrungsbericht und Zeiterfassung nachführen |
| Feinplanung nächste Phase |
| **Konzeption 1** | Software Architektur Dokument |
| Externe Libraries überprüfen |
| Aufbau der Dokumentation |
| Autokorrelation definieren |
| MersenneTwister definieren |
| Unterschied der verschiedenen Distributionen aufzeigen |
| Cholesky Decomposition dokumentieren |
| Lösung für Checked Exceptions definieren |
| Aufbau und Änderungen der C# Library dokumentieren |
| Testing definieren |
| Erfahrungsbericht und Zeiterfassung nachführen |
| Feinplanung nächster Phase |
| **Konzeption 2** |  |
| **Implementation** | Planung der Implementation |
| Implementation: JARTA in c# |
| Implementation: Integration ARTA # in Simio |
| Implementation: Testing |
| Code Reviews |
| Erfahrungsbericht und Zeiterfassung nachführen |
| Feinplanung nächste Phase |
| **Testing** | Wie und Was wird getestet? |
| Testen implementieren |
| Korrekturen an der Implementierung |
| Testen in Simio |
| Generierte Daten mit Jarta vergleichen |
| Dokumentieren |
| Erfahrungsbericht und Zeiterfassung nachführen |
| Feinplanung nächste Phase |
| **Simulation & Auswertung** | Simulationsumgebung aufsetzen |
| Simio Videos erstellen |
| Simulation & Datensammlung |
| Dokumentation |
| Fertigstellung |
|  | Erfahrungsbericht und Zeiterfassung nachführen |
|  | Feinplanung nächste Phase |

# 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 |
| Literaturverzeichnis | Citavi |

# 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 Excel Liste (siehe Referenzen), da Zube diese Funktionalität nicht mit sich bringt.

In der Zeiterfassung sind, ein Total der Arbeitsstunden und die Sprints mit den jeweiligen Arbeitspaketen aufgeführt.

## Entwicklung

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

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

### Code Reviews & Unit Testing

Code Reviews werden vor Abschluss der jeweiligen Arbeitspakete zu zweit ausgeführt, mit dem Ziel, die Anforderungen von Herrn Kretschmar zu erfüllen.

Zudem werden Code Reviews mit Herrn Kretschmar durchgeführt um von seinem Know-How zu profitieren und unseren Code zu verbessern.

Unit-Tests werden im Rahmen der Implementation zur Überprüfung der mathematischen Korrektheit des ARTA-Prozesses erzeugt.

Zur Überprüfung der numerischen Korrektheit werden statistische Tests im Code realisiert.

# Simulation

Eine Simulationsumgebung wird in der Phase «Simulation» erarbeitet. Ziel dieser Umgebung ist es, die erzeugte Klassenbibliothek auf eine saubere Integration ins Simio zu testen. Weiter wollen wir Daten erzeugen, um die Beobachtung der abweichenden Resultate zwischen autokorrelierten Zufallszahlen und solchen die keine Zusammenhänge aufweisen zu bestätigen.