

Technische Universität Dortmund
Fakultät Maschinenbau
Fachgebiet IT in Produktion und Logistik
Univ.-Prof. Dr. Ing. Markus Rabe

Fachwissenschaftliche Projektarbeit

Geschäftsprozessoptimierung unter Nutzung von Parametertuning

Autor: Mert Koymatli

Matrikelnummer: 204518

Ausgegeben am: 14.04.2021

Eingereicht am: 14.10.2021

Betreuer: Dr.- Ing Dipl.-Inform. Anne Antonia Scheidler, M. Sc. Florian Hochkamp

Inhaltsverzeichnis

Abbildungsverzeichnis.....	I
Tabellenverzeichnis.....	II
Abkürzungsverzeichnis	III
1. Einleitung.....	1
2. Grundlagen.....	3
2.1 Geschäftsprozess	3
2.2 Optimierung unter Nutzung von Parametertuning	6
2.3 Geschäftsprozessoptimierung.....	7
2.4 Simulation	8
3. Modellierung eines Geschäftsprozesses	11
3.1 Anwendungsbeispiel	11
3.2 Modellierung mit einer ereignisgesteuerten Prozesskette.....	12
4. Simulative Untersuchung.....	16
4.1 Implementierung in Arena	16
4.2 Experimentplanung	18
4.3 Simulationsexperiment.....	19
4.4 Verifikation & Validierung	21
4.5 Diskussion & Fazit.....	22
5. Zusammenfassung und Ausblick.....	25
Literaturverzeichnis	26
Anhang.....	28

Abbildungsverzeichnis

Abbildung 1: Basis Elemente der EPK	S.6
Abbildung 2: Skizze der Fertigungslinie	S.11
Abbildung 3: Bearbeitungsstation als Funktion.....	S.13
Abbildung 4: EPK des Teilprozesses	S.15

Tabellenverzeichnis

Tabelle 1: eEPK Elemente	S. 6
Tabelle 2: Parameterwerte des Anwendungsbeispiels.....	S. 12
Tabelle 3: Pausenplanung.....	S.17
Tabelle 4: Variation der Pausenplanung	S. 20
Tabelle 5: Variation der Priorisierung	S. 20
Tabelle 6: Optimale Parametereinstellung	S. 21
Tabelle 7: Grenzwerttest	S.22

Abkürzungsverzeichnis

GP	Geschäftsprozess
GPO	Geschäftsprozessoptimierung
UML	Unified Modeling Language
BPMN	Business Process Modeling Notation
EPK	Ereignisgesteuerte Prozessketten
V&V	Verifikation und Validierung
FIFO	First In - First Out
LIFO	Last In - First Out
DV	Dreiecksverteilung
DLZ	Durchlaufzeit
WZ	Wartezeit

1. Einleitung

Unternehmen müssen sich ständig mit der eigenen Marktposition auseinandersetzen (Wicharz, 2018). Regelmäßige Weiterentwicklungen können vorteilhaft sein und die Konkurrenzfähigkeit steigern (Wicharz, 2018). Eine Möglichkeit, die Leistungsfähigkeit zu steigern, ist die Optimierung von Geschäftsprozessen (GP). Gadatsch begründet das Bestreben nach Optimierung mit dem vorhandenen Kostendruck der Unternehmen (Gadatsch, 2010). Infolgedessen wird deutlich, dass für Unternehmen die Geschäftsprozessoptimierung (GPO), welche insbesondere in den 90er-Jahren durch Werke von Pionieren wie z. B. Hammer und Champy (1995) an Bedeutung gewann, für Unternehmen ein attraktives Verfahren darstellen kann, um sich weiterzuentwickeln (Becker et al. 2009). Allgemein kann die GPO zur Leistungs- und Qualitätssteigerung und somit zu einer besseren Marktposition führen (Becker et al., 2009).

In der Literatur finden sich viele verschiedene Optimierungsverfahren, die sich im Rahmen der GPO bewähren (Gadatsch, 2010). Ein Optimierungsverfahren, welches nicht mit der GPO in Verbindung gebracht wird, ist das Parametertuning. Ziel dessen ist das Finden von optimalen Parametern bzw. Parameterkombinationen. Die GPO unter Nutzung von Parametertuning könnte verschiedene Vorteile mit sich bringen. Beispielsweise könnten GP mit nur geringem Aufwand hinreichend optimiert werden, weil keine komplexe Umstrukturierung der GP nötig wäre.

Ziel der Arbeit ist es zu untersuchen, wie sich das Parametertuning mit der GPO verbinden lässt. Hierbei soll das manuelle Parametertuning auf eine Produktionsumgebung mit einem GP angewendet werden. Mit Hilfe der Simulationssoftware „Arena“ von Rockwell Automation soll hierzu ein GP modelliert werden. Durch ein Simulationsexperiment soll geprüft werden, wie das Parametertuning zur GPO genutzt werden kann und welche Herausforderungen sich daraus ergeben. Darüber hinaus soll die Eignung des manuellen Parametertunings zur GPO diskutiert werden.

Zu Beginn der Arbeit wird eine geeignete Definition für den GP gesucht und eine allgemeine Erklärung zu dem Begriff der Optimierung geliefert. Letzteres wird mit dem Parametertuning in Verbindung gesetzt. Auf der Basis dieser Definition wird der Begriff GPO definiert und eingeordnet. Des Weiteren werden die Modellierungs-

und Simulationsansätze die im Rahmen dieser Arbeit verwendet werden vorgestellt. Im dritten Kapitel wird zuerst ein GP vorgestellt welcher anschließend in Form einer ereignisgesteuerten Prozesskette (EPK) modelliert wird. Der GP beruht auf einem GP welcher von Gutenschwager et al. (2017) vorgestellt worden ist. Es handelt es sich um eine Produktionslinie für Industrie-PCs. Das Modell soll als Grundlage für die darauffolgende Implementierung in Arena sein. Mit dem Simulationsmodell soll untersucht werden, ob und wie eine GPO unter Nutzung von Parameter tuning möglich ist. Der GP soll dann als optimiert gelten, wenn der Output gesteigert und die durchschnittliche Durchlaufzeit (DLZ) der Werkstücke gesenkt werden kann. Anschließend soll das Simulationsexperiment verifiziert und validiert werden. Dazu werden der Grenzwerttest und der Kontinuitätstest verwendet. Daraufhin wird auf der Grundlage dieser Untersuchung diskutiert, inwieweit sich das Parameter tuning mit der GPO verbinden lässt. Zum Abschluss werden die Arbeit und ihre Ergebnisse zusammengefasst und ein Forschungsausblick gegeben.

2. Grundlagen

In diesem Kapitel werden die grundlegenden Begriffe dieser Arbeit definiert. Außerdem werden die im späteren Verlauf der Arbeit verwendeten Modellierungs- und Simulationsansätze erläutert. Abschließend soll die Simulationssoftware Arena beschrieben werden. Das Kapitel soll dazu dienen, die Grundlagen zu schaffen, die für diese Arbeit notwendig sind.

2.1 Geschäftsprozess

Der Begriff GP bildet die Grundlage für diese Projektarbeit und stellt den zentralen Untersuchungsgegenstand dar. Bevor nähere Untersuchungen durchgeführt werden, ist es notwendig, den Begriff GP näher zu erläutern.

In der Literatur lassen sich viele verschiedene Ansätze zur Definition des Begriffs Geschäftsprozess finden (Schwarz et al., 2018). Die verschiedenen Definitionsansätze unterscheiden sich überwiegend in der Perspektive und Präzision, wenngleich sie alle dasselbe zu beschreiben versuchen. Jacobson erklärt dieses Phänomen mit der unsichtbaren Natur von Geschäftsprozessen (Lindsay et al., 2003).

Rump definiert GP als „eine zeitlich und sachlogisch abhängige Menge von Unternehmensaktivitäten, die ein bestimmtes, unternehmensrelevantes Ziel verfolgen und zur Bearbeitung auf Unternehmensressourcen zurückgreifen.“ (Rump 1999, S. 19 zitiert nach Staud 2006, S. 8) Eine alternative Definition wird von Hammer und Champy gegeben. Sie beschreiben einen GP als einen „Bündel von Aktivitäten, für das ein oder mehrere unterschiedliche Inputs benötigt werden und das für den Kunden ein Ergebnis von Wert erzeugt.“ (Hammer & Champy, 1995, S. 52 zitiert nach Staud 2006, S. 9)

Beide Definitionen eignen sich gut als eine erste Einordnung des Begriffs. Nichtsdestotrotz lohnt es sich diese Definitionen weiter auszuführen, um einen genaueren Einblick in die Bedeutung von GP zu bekommen. Palleduhn und Neuendorf (2013), Schwarz et al. (2018) und Staud (2006) fassen die zentralen Punkte verschiedener Definitionen zusammen, um eine vollständigere Begriffsdefinition zu formulieren.

Staud weist darauf hin, dass in der Literatur häufig der Fehler gemacht wird Wünsche an einen optimalen GP als Definition darzustellen (Staud, 2006). Jedoch erfüllt nicht zwangsmäßig jeder GP, die an Ihn gestellten Wünsche.

Des Weiteren wird der Begriff des GP in der Literatur vom Workflow abgegrenzt. Österle beschreibt den Workflow „als einen verfeinerten Geschäftsprozess“ (Gadatsch 2010, S. 65). Workflows unterscheiden sich grundsätzlich durch ihren hohen Detaillierungsgrad von GP. Die Aufgaben in einem Workflow sind so detailliert, sodass sie als Arbeitsanweisungen für die Mitarbeiter genutzt werden können (Gadatsch 2010, S. 65).

Ein von Hammer und Champy ursprünglich eingeführter Gedanke ist, dass GP ein oder mehrere Inputs in Outputs umwandeln. Nach Palleduhn und Neuenbauer handelt es sich bei den Inputs sowie Outputs um Unternehmensressourcen in Form von Materie, Energie und Information (Palleduhn & Neuendorf, 2013). Der Output eines Prozesses kann als Input für einen folgenden Prozess genutzt werden (Becker et al., 2009). Ein Geschäftsprozess beginnt und endet bei einem Kunden (Palleduhn & Neuendorf, 2013). Ob diese Kunden unternehmensextern oder -intern sind, ist hierbei irrelevant (Rosenkranz, 2006). Die Länge eines GP ist wiederum vollständig subjektiver Natur und kann objektiv nicht festgelegt werden (Staud, 2006). GP können als ereignisdiskret beschrieben werden (Palleduhn & Neuendorf, 2013). Folglich wird jeder einzelne Schritt eines GP durch ein Ereignis angeregt (Palleduhn & Neuendorf, 2013). In der Regel werden GP arbeitsteilig ausgeführt (Palleduhn & Neuendorf, 2013). Diese Arbeitsteilung kann abteilungs-, aber auch unternehmensübergreifend erfolgen (Palleduhn & Neuendorf, 2013). Hierbei ist auch eine Miteinbeziehung verschiedener Hierarchieebenen üblich (Staud, 2006). GP werden meist mehrfach wiederholt (Palleduhn & Neuendorf, 2013).

Palleduhn und Neuendorf fassen ihre Erkenntnisse wie folgt zusammen: „Ein Geschäftsprozess stellt demnach eine strukturierte, zeitlich und logisch zusammengehörige Abfolge von Unternehmensaktivitäten mit klar definierten Anfangs- und Endpunkten dar. Er kombiniert Ressourcen und zu bearbeitende Objekte als Input mit dem Ziel, eine bestimmte Leistung als Output zu erstellen“ (Palleduhn & Neuendorf, 2013, S. 4). Diese Definition soll in dieser Arbeit verwendet werden, da sie die wichtigsten Aussagen verschiedener Definitionen zusammenfasst.

In der Regel ist es von Vorteil, GP grafisch darzustellen bzw. zu modellieren. Dadurch können GP die in der Realität komplex sind, auf das wesentlichste Beschränkt werden und eine geeignete Übersicht geschaffen werden.

„Das Verhalten des realen Systems zu beschreiben“ ist laut Becker die Aufgabe der Prozessmodellierung (Becker et al., 2009, S. 35). Hierbei ist darauf zu achten, dass es keine allgemeingültige Modellierungsmethode gibt (Becker et al., 2009). Es existieren viele verschiedene Variationen, die verschiedene Vor- und Nachteile mit sich bringen. Jedoch muss dem Modellierer stets bewusst sein, dass Modelle die Realität nie vollständig abbilden können und immer nur ein vereinfachtes Bild dieser darstellen wird (Gutenschwager et al., 2017).

In der Literatur existieren viele verschiedene Modellierungsmethoden. Bekannt ist beispielsweise die Unified Modeling Language (UML). UML-Diagramme werden häufig in der objektorientierten Software-Entwicklung genutzt. Eine für die Prozessmodellierung genutzte Methode ist die Business Process Modeling Notation (BPMN). Bei der BPMN werden hierzu eine Vielzahl an Beschreibungselementen genutzt. Diese unterstützen zwar die Korrektheit und Präzision der Beschreibung, jedoch leidet die Verständlichkeit unter dieser Komplexität (Obermeier et al 2014).

Eine sehr weit verbreitete Modellierungsmethode ist die der EPK (Becker et al., 2009). Becker beschreibt EPK wie folgt: „Die ereignisgesteuerte Prozesskette ist eine semi-formale Modellierungssprache und dient zur fachkonzeptionellen Modellierung von Prozessen. Durch das Aneinanderreihen von Funktionen und Ereignissen lassen sich komplexe Abläufe bilden, die zusammengenommen die Abfolge der Funktionen zur Bearbeitung eines betriebswirtschaftlichen Objektes darstellen“ (Becker et al., 2009, S. 43–44) . EPK bauen auf Petri-Netzen, die große Bekanntheit in der Automatentheorie erreicht haben, auf (Becker et al., 2009, S. 43).

Aufgrund der Schlichtheit und einfachen Verständlichkeit dieser Modellierungsmethode, soll diese im Rahmen der Arbeit genutzt werden. EPK bestehen im Wesentlichen aus Ereignissen und Funktionen, die durch einen Kontrollfluss miteinander verbunden werden. Ein Ereignis ist die Beschreibung eines vorliegenden Zustands. Das Ereignis ist ein passives Element, das lediglich eine Funktion auslösen kann. Prozesse starten und Enden mit Ereignissen. Eine Funktion wiederum beschreibt die Umwandlung eines Inputs- in ein Output. Dadurch können Sie als aktives Element beschrieben werden. Auf ein Ereignis muss eine Funktion folgen und auf eine Funktion muss ein Ereignis folgen. Kontrollflüsse verbinden diese Elemente. Sie stellen den Zeitfluss in der Prozesskette dar. Konnektoren können Kontrollflüsse teilen und zusammenführen. Diese sind das „Und“, das „inklusive Oder“ und das

„exklusive Oder“. In Abbildung 1 ist eine mögliche grafische Darstellung dieser Basis Elemente von EPK dargestellt (Becker et al., 2009, S.43-57).

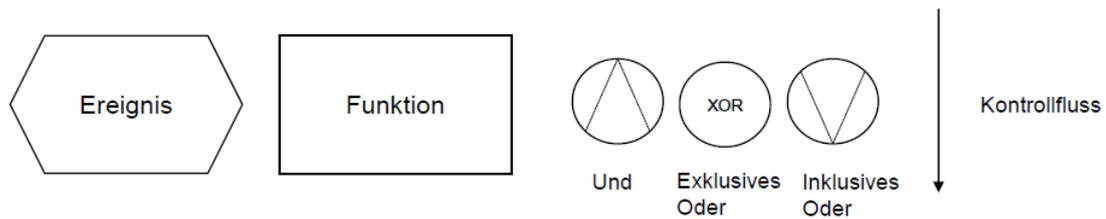


Abbildung 1: Basis Elemente der EPK (In Anlehnung an Becker et al., 2009, S.47)

EPK können um zusätzliche Elemente erweitert werden. Bei diesen Elementen handelt es sich um Informationsobjekte, Organisationseinheiten und Ressourcen, welche an Funktionen angehängt werden können. Sie können dazu genutzt werden, um EPK detaillierter zu gestalten und den realen Prozess besser zu beschreiben (Becker et al., 2009). In Tabelle 1 sind die für diese Arbeit relevanten Elemente aufgelistet.

Element	Bedeutung
	Organisationseinheit Welche Person (Personenkreis) führt eine Funktion
	Nicht gerichtete Zuordnungskette

Tabelle 1: eEPK Elemente (Tabelle angelehnt an Becker et al., 2009, S.53)

2.2 Optimierung unter Nutzung von Parametertuning

Ein weiterer fundamentaler Begriff dieser Projektarbeit ist der Begriff des Parametertunings. Wie in der Einleitung hingewiesen, soll überprüft werden, ob und inwiefern sich GP mit dem Parametertuning optimieren lassen. Hierzu soll zunächst geschaut werden, was die Optimierung im Allgemeinen ist und was das Parametertuning auszeichnet.

Es existieren verschiedene Möglichkeiten den Begriff der Optimierung zu erläutern. Grimme beschreibt die Optimierung als „das Bemühen um Verbesserung von Werkzeugen, Prozessen, Organisationsformen oder auch vollständigen Systemen [...]“. (Grimme & Bossek, 2018). Diese kurze Definition von Grimme ist zweckdienlich für diese Arbeit.

Ein Optimierungsverfahren, das in der Mathematik häufig angewandt wird, ist das Parametertuning. Das Ziel des Parametertunings ist es, die optimalen Parameter, sowie die optimale Kombination dieser zu finden, um ein gegebenes Ziel zu erreichen.

Es ist zu beachten, dass das Parametertuning von der Parameterkontrolle abgegrenzt werden muss. Beim Parametertuning werden Parameterkombinationen vor einem Simulationslauf bestimmt (Huang et al, 2020). Bei der Parameterkontrolle hingegen ändern sich die Parameter während des Simulationslaufs (Huang et al., 2020).

Es existieren verschiedene Arten Parametertuning durchzuführen. Parametertuning kann sowohl manuell als auch automatisch durchgeführt werden. Automatisches Parametertuning beruht auf Parametertuningalgorithmen. Jedoch muss beachtet werden, dass nicht jeder Algorithmus für jedes Optimierungsproblem geeignet ist. Algorithmen sollten idealerweise an das Problem oder einer Klasse von Problemen angepasst werden. Dies kann dazu führen, dass vor allem bei vielen verschiedenen Optimierungsproblemen eine hohe Expertise des Ausführenden gefordert wird. (Huang et al., 2020)

Ein simplerer Ansatz ist der Ansatz des manuellen Parametertunings. Dieser wird seit langer Zeit benutzt und findet in der Praxis immer noch Anwendung. Beim manuellen Parametertuning werden die Parameter durch ein iteratives Vorgehen durch die Entscheidungen des Anwenders optimiert (Elizabeth Montero et al. 2014). Das manuelle Parametertuning kann mit Hilfe eines Experimentplans und eines Simulationsexperiments durchgeführt werden.

2.3 Geschäftsprozessoptimierung

Wenn man versucht, verschiedene Optimierungsverfahren für GP einzuordnen, stößt man schnell auf den Begriff der GPO. In dem Rahmen der GPO, soll untersucht werden, ob GP und das Optimieren unter Nutzung von Parametertuning sich miteinander verbinden lässt. Versucht man eine geeignete Definition für die GPO zu finden, fällt auf das auch dieser Begriff in der Literatur nicht eindeutig definiert ist. Bei der Definition des GP sind zwar verschiedene Definitionsansätze vorhanden, jedoch sind die Bedeutungen des Worts GP stets nah beieinander. Dies erweist sich bei der Definition der GPO als problematischer. Zwar bleibt die triviale Bedeutung,

dass es sich um die Optimierung von GP handelt, bei allen Autoren unverändert, jedoch unterscheiden sich die Abgrenzungen des Begriffs in der Literatur enorm. Einige verwenden den Begriff der GPO als Oberbegriff für alle Arten von Optimierungsmethoden von Geschäftsprozessen in Unternehmen, für andere ist es wiederum eine spezifische Optimierungsmethode.

Im Zuge dieser Arbeit soll der Begriff der GPO klar von anderen Optimierungsmethoden abgegrenzt werden und als eigene Methode angesehen werden. Kaeseler kritisiert in seiner Arbeit, dass die Begriffe Business Process Reengineering (BPR) und GPO häufig als synonym verwendet werden (Kaeseler, 1996). Er unterscheidet klar zwischen beiden Optimierungsverfahren. Während BPR eine tiefgreifende und radikale Umstrukturierung der Unternehmensstruktur bedeutet, beschäftigt sich die GPO mit der „Optimierung von Bestehendem“ (Kaeseler, 1996, S. 6). Gadatsch weist darauf hin, dass aufgrund der radikalen Veränderungen beim BPR, Chancen und Risiken sehr hoch sind (Gadatsch, 2010). GPOs hingegen, aufgrund der kleinen und moderaten Veränderung, weniger riskant sind (Gadatsch, 2010).

Die Notwendigkeit solcher Optimierungsmethoden wird dann deutlich, wenn man sich die Entstehung von GP vor Augen führt. Laguna und Marklund führen an, dass GP bei der Entstehung in der Praxis selten aktiv geplant und gestaltet werden. Häufig entstehen GP aus der Not heraus. Beispielsweise kann eine ungeplante Arbeitsteilung bei wachsenden Unternehmen dazu führen, dass GP ineffizient werden. Laguna und Marklund geben das Aufsummieren von lokalen Veränderungen an GP durch Umwelteinflüsse als einen weiteren Grund für ineffiziente GP an (Laguna & Marklund, 2013).

Ob ein GP hinreichend optimiert worden ist, ist eine Frage die nur subjektiv beantwortet werden kann. Dennoch existieren in der Literatur einige Kriterien, die dabei helfen sollen, den Grad der Optimierung zu bestimmen. Typische Bewertungskriterien sind beispielsweise die Zufriedenheit der Kunden, die DLZ, die Kapazitätsauslastung, Fehlerraten etc. (Becker et al., 2009). Innerhalb dieser Arbeit sollen vor allem die DLZ und der Output als Bewertungskriterien herangezogen werden.

2.4 Simulation

Um zu überprüfen, ob es möglich ist GP mit Parametertuning im Rahmen einer GPO zu optimieren, wird in dieser Arbeit ein praktischer Ansatz gewählt. Es soll

versucht werden mit Hilfe einer Geschäftsprozesssimulation festzustellen, ob eine Optimierung des GP durch Parametertuning möglich ist.

Simulation ist ein Begriff, der aus dem lateinischen stammt und in seiner Übersetzung dem deutschen Wort Nachahmung gleichzusetzen ist (Gadatsch, 2010). Gadatsch definiert den Begriff der Simulation als „die Nachbildung der Realität in einem Modell, um damit zu experimentieren“ (Gadatsch, 2010, S. 216). Die Simulation hat sich vor allem zur Analyse von Produktionsprozessen etabliert und unterstützt Unternehmen bei ihren Entscheidungen (Gadatsch, 2010). Gadatsch vergleicht die Simulation mit einem Probetrieb, der mit Hilfe des Computers stattfindet und ohne Konsequenzen ausgeführt werden kann (Gadatsch, 2010). Er sieht in Simulationsexperimenten eine Möglichkeit, beispielsweise Warte- und Liegezeiten, die häufig einen großen Teil der DLZ ausmacht, zu reduzieren (Gadatsch, 2010). Bei der Simulation werden Eingangsgrößen in Ausgangsgrößen umgewandelt. Hierbei werden die Beziehungen zwischen Ein- und Ausgangsgrößen durch das Simulationsmodell festgelegt. Eingangsgrößen sind entweder variabel oder werden als feste Parameter vorgegeben. (Gadatsch, 2010)

Ziel ist es ein experimentierfähiges Modell zu erzeugen. „Als Simulationsexperiment wird nach VDI 2206 die gezielte empirische Untersuchung des Modellverhaltens auf der Basis wiederholter Simulationsläufe bezeichnet, wobei das Simulationsmodell systematisch hinsichtlich seiner Parameter oder seiner Struktur variiert werden kann“ (Gutenschwager et al., 2017, S. 23). Gutenschwager beschreibt den Simulationslauf als „die Ausführung des Simulationsmodells mit einer festgelegten Parameterkonfiguration über ein vorgegebenes Zeitintervall.“(Gutenschwager et al., 2017, S. 23). Des Weiteren führt er aus, dass ein Simulationslauf bei gleicher Parametereinstellung i. d. R. reproduzierbar ist (Gutenschwager et al., 2017, S. 23).

Eine geeignete Möglichkeit ein Simulationsexperiment durchzuführen, besteht darin, die zu untersuchenden Parameterkonfiguration vorher in einer Tabelle festzuhalten und später um die Ergebnisse zu ergänzen. Eine solche Tabelle wird auch Experimentplan genannt. Ein Experiment kann auch aus mehreren solcher Experimentplänen bestehen (Gutenschwager et al., 2017). In der Literatur finden sich verschiedene Ansätze zur Erstellung von Experimentplänen. Ein geläufiger Ansatz ist der Ein-Faktor-Ansatz. Hierbei gilt, dass jeweils immer nur ein einziger Parameter

variiert werden soll. Dies soll dabei helfen, den Einfluss der einzelnen Parameter auf das Modellverhalten zu erkennen (Gutenschwager et al., 2017).

Um eine Simulation bzw. ein Simulationsexperiment durchzuführen, wird eine Simulationssoftware benötigt. Zu diesem Zweck wird in dieser Arbeit die Simulationssoftware Arena von Rockwell Automation genutzt.

Die Firma Rockwell Automation bietet eine kostenlose Studentenversion ihrer Simulationssoftware Arena an (Rockwell Automation, 2021). Diese ist zwar in ihrer Modellgröße und ihrer maximalen Anzahl an Entitäten begrenzt, reicht aber für die Zwecke dieser Arbeit vollständig aus. Für diese Projektarbeit wird die Version 16.10.00001 der Studentenversion von Arena genutzt.

Jedes erzeugte und verwendete Simulationsmodell sollte anschließend oder während der Erzeugung mittels Verifikation und Validierung (V&V) geprüft werden. In der Literatur finden sich viele verschiedene V&V-Techniken, die hierzu verwendet werden können. In dieser Arbeit werden der Kontinuitätstest und der Grenzwerttest verwendet werden. Diese eignen sich gut für Simulationsmodelle und können ohne großen Aufwand zu Prüfung der V&V genutzt werden. Beim Grenzwerttest wird überprüft, ob sich das Modell auch für Extremwerte, sinnvoll verhält (Rabe et al., 2008). Bei der Anwendung des Grenzwerttestes kann es hierbei hilfreich sein, Parameterwerte so zu setzen, sodass die Vorhersagbarkeit des Modells steigt. Unter anderem ist es möglich diese so zu wählen, dass bestimmte Abschnitte des Modells nicht mehr durchlaufen werden. Beispielsweise kann die Wahrscheinlichkeit für die Rückführung eines fehlerhaften Bauteils in einer Produktionslinie gleich null gesetzt werden (Rabe et al., 2008).

Eine weitere V&V-Technik stellt der Kontinuitätstest dar. Hierbei wird untersucht ob geringe Veränderung des Inputs zu verhältnismäßig geringer Änderung des Outputs resultieren, es sei denn, sie können durch den zugrunde liegenden Prozess verstanden und begründet werden (Shannon 1981). Große Schwankungen und Ausreißer können somit auf Fehler des Modells hinweisen. Der Kontinuitätstest kann gut mit einer exemplarischen Durchführung verbunden werden.

3. Modellierung eines Geschäftsprozesses

In diesem Kapitel soll der zu behandelnde GP vorgestellt werden. Um einen geeigneten Übergang zwischen dem Fallbeispiel und der Implementierung als Simulationsmodell zu schaffen, soll der GP anschließend in Form einer EPK modelliert werden.

3.1 Anwendungsbeispiel

Der in dieser Arbeit verwendete GP ist angelehnt an den von Gutenschwager et al. (2017) beschriebenen Prozess. Bei diesem Beispiel handelt es sich um die Produktion von Industrie-PCs.

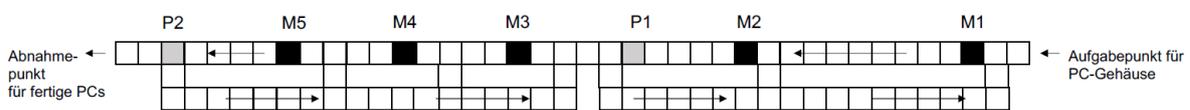


Abbildung 2: Skizze der Fertigungslinie (In Anlehnung an Gutenschwager et al., 2017, S.8)

In dieser Fertigungslinie, die in Abbildung 2 grob skizziert ist, sollen PC-Gehäuse angeliefert werden, welche über die fünf Bearbeitungsstationen M1 bis M5 zu einem Industrie-PC gefertigt werden. Die Wahrscheinlichkeit einer fehlerhaften Bearbeitung an seiner Bearbeitungsstation soll 5% betragen. Innerhalb der Fertigungslinie sind zwei Prüfstationen P1 und P2 vorhanden. An diesen soll geprüft werden, ob die einzelnen Werkstücke ordnungsgemäß bearbeitet worden sind. Ist dies nicht der Fall, sollen die Werkstücke an die Bearbeitungsstation, an welcher der Fehler entstanden ist, zurückgeführt werden. Reparaturzeiten an den Bearbeitungsstationen unterscheiden sich von den ordnungsgemäßen Bearbeitungszeiten. Gilt ein Werkstück als repariert, durchläuft es die nachfolgenden Stationen wie ein ordnungsgemäß bearbeitetes Werkstück.

Die Werkstücke werden zwischen den Stationen durch Förderbänder transportiert. Transportzeiten können hierbei vernachlässigt werden. Vor allen Bearbeitungs- und Prüfstationen sind Puffer als Teil der Förderbänder vorhanden. Diese sollen vereinfachend nicht in ihrer Kapazität begrenzt werden. Dennoch soll darauf geachtet werden, größere Anstauungen an diesen zu vermeiden. Nach Gutenschwager et al. (2017) sollen die Puffer nach dem First In-First Out-Prinzip (FIFO) arbeiten.

Zusätzlich zu den Beschreibungen von Gutenschwager et al. (2017) wird angenommen, dass jede Station einen Mitarbeiter benötigt. Es wird angenommen, dass die Produktivität der Mitarbeiter über den Zeitverlauf konstant bleibt und kein

Unterschied zwischen den Mitarbeitern existiert. Dennoch ist jeder Mitarbeiter lediglich zu Arbeiten an seiner eigenen Arbeitsstation befähigt.

Die Mitarbeiter müssen innerhalb ihrer 8 Stunden Schicht, die gesetzlich vorgeschriebene Pause von einer halben Stunde wahrnehmen. Während dieser Pause, pausieren auch die Arbeiten an der jeweiligen Station. Zunächst soll angenommen werden, dass alle Mitarbeiter nach 4 Stunden gleichzeitig eine 30-minütige Pause machen. Aufgrund der geringen Bearbeitungszeiten pro Werkstück verlassen die Mitarbeiter ihren Arbeitsplatz zur Pause erst nachdem begonnene Arbeiten an einer Station beendet worden sind.

Vereinfachend soll angenommen werden, dass die Stationen die benötigten Werkzeuge und Materialien stets vorliegen haben, wodurch eine Betrachtung der logistischen Prozesse um den GP herum entfallen kann.

Parameter wie die Taktzeit, Bearbeitungszeit usw. sollen dem Buch von Gutenchwager et al. (2017) entnommen werden. Werte, für die keine genauen Angaben zu finden sind, sollen sinnvoll ergänzt werden. Die verwendeten Parameter sind der folgenden Tabelle 2 zu entnehmen. Anzumerken ist noch, dass die Bearbeitungszeiten einer Dreiecksverteilung (DV) gleichen. Dadurch soll dargestellt werden, dass Menschen nie exakt gleich schnell arbeiten können.

Parameter	Verteilung	Mittelwert in s	Minimum in s	Maximum in s
Taktzeit	-	220	-	-
Bearbeitungszeit M1	DV	219	199	239
Bearbeitungszeit M2	DV	215	195	235
Bearbeitungszeit M3	DV	210	190	230
Bearbeitungszeit M4	DV	214	194	234
Bearbeitungszeit M5	DV	215	195	235
Prüfzeit P1	DV	60	40	80
Prüfzeit P2	DV	60	40	80
Reparaturzeit	DV	100	60	140

Tabelle 2: Parameterwerte des Anwendungsbeispiel

3.2 Modellierung mit einer ereignisgesteuerten Prozesskette

Anstatt direkt ein Simulationsmodell in Arena zu erzeugen, wird zunächst ein Modellentwurf in Form einer EPK erzeugt. Dies hilft dabei, eine Übersicht zu erzeugen und einen komplexen GP aus der Realität auf das Wesentliche zu beschränken.

Da der betrachtete GP lang ist und die Länge im Wesentlichen durch sich wiederholende Elemente zustande kommt, sollen sinnvolle Zusammenfassungen von Prozessschritten das Modell vereinfachen. Die Prozessschritte an einer Bearbeitungs- und Prüfstation bleiben stets gleich. Durch die Zusammenfassung der Prozessschritte als eigene Funktionen kann das Modell verkürzt werden. In der folgenden Abbildung 3 wird die Zusammenfassung der Prozessschritte einer niedrigeren Ebene zu einer Funktion höherer Ebene dargestellt.

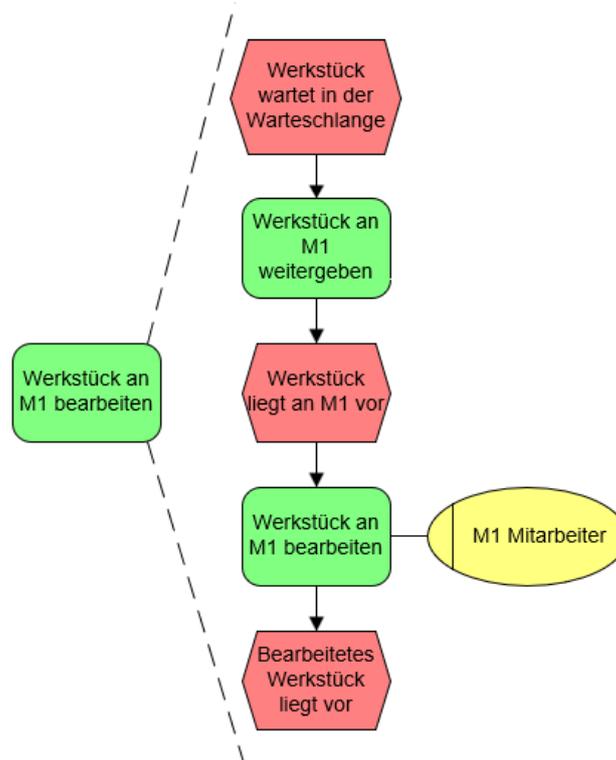


Abbildung 3: Bearbeitungsstation als Funktion

Die Zusammenfassung an einer Prüfstation funktioniert analog (siehe Anhang 1). Mit diesen Vereinfachungen verkürzt sich das zu erzeugende Modell. Das Modell muss nach den Regeln von EPK mit einem Ereignis starten. In dem bearbeiteten Anwendungsbeispiel ist das Eintreffen des PC-Gehäuses (bzw. Werkstücks) ein geeignetes Startereignis. Nach dem Eintreffen des Werkstücks wird dieses in die Warteschlange bzw. den Puffer vor der Bearbeitungsstation M1 einsortiert. Daraus resultiert das Ereignis „Werkstück wartet in der Warteschlange“. Wenn die Bearbeitungsstation M1 leer ist, d. h. kein anderes Werkstück an dieser bearbeitet wird, wird das Werkstück an die Station M1 weitergegeben und an dieser bearbeitet. Dabei kann durch eine Organisationseinheit (vgl. Tabelle 1) dargestellt werden, dass für die Bearbeitung des Bauteils an der Station M1 der dazugehörige Mitarbeiter notwendig ist. Die gleichen Prozessschritte werden analog für das Bauteil in der

Bearbeitungsstation 2 durchgeführt. Nach der Funktion „Bauteil an M2 bearbeiten“ liegt ein bearbeitetes Bauteil vor, welches in den nächsten Schritten geprüft werden muss. Dazu wird das Bauteil in den Puffer vor P2 einsortiert und sobald möglich, an die Prüfstation P1 weitergegeben. Nach der Prüfung werden fehlerhafte Bauteile zurückgeführt. Um dies mit Hilfe von EPK darzustellen, wird ein „exklusives oder“ genutzt. Die zurückführenden Kontrollflüsse führen zu der jeweiligen Funktion, in der die Bauteile in die Puffer der fehlerverursachenden Bearbeitungsstation einsortiert werden. Falls kein Bauteilfehler festgestellt wird, werden die Bauteile der Bearbeitungsstation M3 weitergegeben. Da der GP sich im Bereich von M3 bis P2 analog zu dem Bereich M1 bis P1 verhält, wird auch das Modell in diesem Bereich analog erzeugt. In Abbildung 4 ist die Modellierung zwischen M1 und P1 als EPK dargestellt. Die vollständige EPK des GP ist im Anhang (siehe Anhang 2) zu finden. Zwar kann durch EPK eine übersichtliche Struktur geschaffen werden, jedoch muss angemerkt werden, dass sie auch einige Nachteile mit sich bringt. Beispielsweise ist das Darstellen von Prozessparametern nicht möglich. Dadurch können Parameter wie die Taktzeit nicht direkt vom Modell abgelesen werden. Auch die Tatsache, dass die Mitarbeiter Pausenzeiten haben und wie diese gestaltet sind, sind nicht ablesbar. Deshalb muss beachtet werden, dass das Modell lediglich als Stütze während der Erzeugung des Simulationsmodells dient und das Modell nicht einfach als Ganzes übernommen werden kann. Durch das Heranziehen von Tabellen, wie z. B. Tabelle 2 kann dieser Nachteil teilweise kompensiert werden.

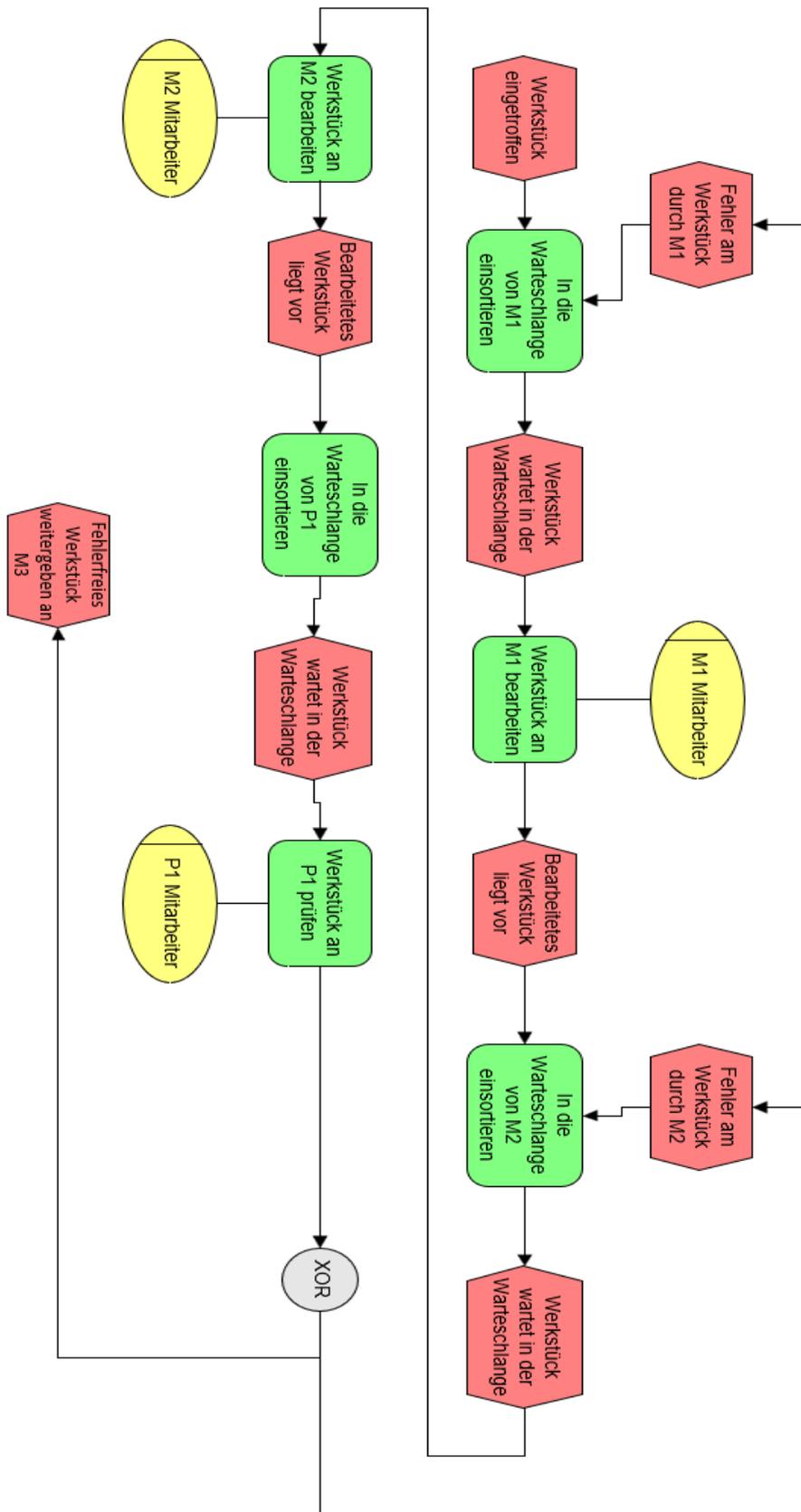


Abbildung 4: EPK des Teilprozesses

4. Simulative Untersuchung

In dem folgenden Kapitel soll der in Kapitel 3.1 vorgestellte GP in ein Simulationsmodell auf der Basis der Simulationssoftware Arena von Rockwell Automation erzeugt werden. Hierzu sollen das Modell aus Kapitel 3.2 und die Tabelle 2 als Hilfe dienen.

4.1 Implementierung in Arena

Wie in der EPK aus Kapitel 3.2 erkennbar, startet der Prozess mit dem Eintreffen eines PC-Gehäuses. Diese PC-Gehäuse werden in Arena durch imaginäre Objekte, welche in der Informatik als Entitäten bezeichnet werden, dargestellt. Für die Erzeugung von Entitäten in Arena nutzt man den Block „Create“. Es sollen Entitäten mit dem Namen „PC-Gehäuse“ erzeugt werden. Die Taktzeit soll konstant 220 Sekunden betragen. Pro Takt soll genau ein PC-Gehäuse erzeugt werden.

Die Bearbeitungs- und Prüfstationen können durch „Process“-Blöcke dargestellt werden. Sie sollen die Entitäten aufnehmen, halten und wieder freigeben. Somit kann die benötigte Zeit für eine Bearbeitung oder Prüfung an der jeweiligen Station simuliert werden. Es muss darauf geachtet werden, dass jeder Station ein eigener Mitarbeiter als „Ressource“ zugewiesen wird, um im späteren Verlauf das Pausieren der einzelnen Stationen bzw. der Mitarbeiter simulieren zu können. Da die Haltezeiten davon abhängen, ob ein Bauteil repariert oder ordnungsgemäß bearbeitet wird, muss davon abgesehen werden, den Stationen feste Haltezeiten zuzuordnen. Ein zielführenderer Ansatz ist es, die Haltezeiten abhängig von den Entitäten zu machen. Dies ist möglich, indem die Haltezeit an einer Station dem Attributwert der jeweiligen Entität gleichsetzt. Durch die Angabe, dass die Stationen die Entitäten aufnehmen, halten und freigeben sollen, erzeugt Arena automatisch eine Warteschlange, die dem jeweiligen „Process“-Block vorangeschaltet ist. Somit entsteht die Warteschlange bzw. der Puffer, der im EPK dargestellt ist automatisch. Die Warteschlange bzw. der Puffer und die Stationen, welche in der Darstellung als EPK noch getrennt vorlagen, werden somit zusammengefasst. Die Priorisierung der Entitäten in der Warteschlange können über den Menüpunkt „Queue“ bestimmt werden.

Wie vorher angedeutet, müssen die „Process“-Blöcke die Entitäten voneinander unterscheiden können. Da zu Beginn des GP alle erzeugten Entitäten gleich sind,

müssen nachträglich Attribute zugewiesen werden. Hierzu wird zu Beginn und nach jeder Bearbeitungs- und Prüfstation den Entitäten die für die nächste Station benötigte Haltezeiten und Priorisierung als Attribut zugeordnet. Diese Zuordnungen sind durch die „Assign“-Blöcke möglich. Die aktualisierten Attribute heißen „Priority“ und „ProcessTime“.

Bei der Betrachtung der vorher erstellten EPK wird deutlich, dass die Rückführungen, die im EPK durch das „exklusive Oder“ erzeugt werden, noch in das Simulationsmodell überführt werden müssen. In Arena kann zu diesem Zweck der „Decide“-Block genutzt werden. Da die Wahrscheinlichkeit, dass an einer Bearbeitungsstation ein Bauteil fehlerhaft bearbeitet wird bei 5 % liegt, kann der Typ des „Decide“-Blocks auf „N-way-by-chance“ gestellt werden. Somit ist es in Abhängigkeit von Wahrscheinlichkeiten möglich, dass Werkstücke als fehlerhaft erkannt werden und zur fehlerverursachten Bearbeitungsstation zurückgeführt werden. Die an der Prüfstation P2 als fehlerfrei erkannten Industrie -PCs werden an werden an einen Abgabepunkt transportiert. Dies kann durch den „Dispose“-Block dargestellt werden. Er stellt eine Senke dar, in der Entitäten aufgenommen werden. Das vollständige Simulationsmodell kann dem Anhang entnommen werden (siehe Anhang 3)

Eine Problematik, die im Kapitel 3.2 bei der Modellbildung erwähnt worden ist, ist, dass man beispielsweise die Pausenzeiten und ihre Gestaltung in der EPK nicht erkennen kann. Auch bei dem in Arena erstellten Simulationsmodell ist dies nicht erkennbar. Wie vorher angedeutet ist hierfür die Zuordnung von Ressourcen in Form von Mitarbeitern zu den Stationen wichtig. Eine Station führt seine Aufgaben nur aus, wenn alle benötigten Ressourcen (also der jeweilige Mitarbeiter) vorhanden sind. Arena gibt dem Nutzer die Möglichkeit die Anzahl der Ressourcen über die Zeit zu variieren. Hierzu können „Schedules“ genutzt werden. Sie geben an, zu welchem Zeitpunkt, wie viele Ressourcen einer Art vorhanden sein sollen. Für den Fall, dass alle Mitarbeiter zur gleichen Zeit Pause machen, wird der in Tabelle 3 „Schedule“ von jeder Ressource genutzt.

Reihenfolge	Aktive Mitarbeiteranzahl	Dauer in h
1	1	4
2	0	0,5
3	1	3,5

Tabelle 3: Pausenplanung

4.2 Experimentplanung

Bevor ein Simulationsexperiment mit dem Simulationsmodell in Arena durchgeführt werden kann, muss konkretisiert werden, welche Parameter durch das Parameter-tuning optimiert werden sollen. Außerdem muss festgelegt werden, woran bzw. an welchen Kennwerten eine etwaige Optimierung erkannt werden soll.

Das Ziel des Experiments, soll es sein, festzustellen, ob der durchschnittliche Output innerhalb einer 8-Stunden Schicht erhöht und die durchschnittlichen DLZ gesenkt werden kann. Hierbei soll untersucht werden, ob dies nur durch die Anpassung von Parametern möglich ist. Die Parameter, welche variiert werden sollen, sind die Taktzeit, die Priorisierung in

den Warteschlangen und die Verteilung der Pausenzeiten.

Die Taktzeit gibt an, in welchen zeitlichen Abständen neue PC-Gehäuse an die Fertigungslinie angeliefert werden sowie Entitäten im Modell erzeugt werden. Der Referenzwert beträgt 220 Sekunden. Da es sich bei der Taktzeit um einen stetigen Parameter handelt, sind sehr viele Variationen dieser möglich. Da das Parameter-tuning manuell durchgeführt wird, sind viele Variationen mit einem enormen Zeitaufwand verbunden. Deswegen wird im Rahmen des Experiments der Parameter Taktzeit im Bereich von 150-300s und einem Abstand zwischen den einzelnen Taktzeiten von zehn Sekunden variiert werden. Diese Überlegung begründet sich darin, dass die Bearbeitungszeit der ersten Bearbeitungsstation im Mittel bei 219 Sekunden liegt.

Die Priorisierung innerhalb der Warteschlangen bzw. Puffern ist ein weiterer Parameter, der variiert werden soll. Im Kapitel 3.1 wurde angegeben das diese Warteschlangen nach dem FIFO-Prinzip agieren sollen. Mit Arena ist es möglich Warteschlangen zu simulieren, die nach dem FIFO-Prinzip, dem LIFO-Prinzip oder nach der Priorisierung unterschiedlicher Objekte arbeiten. Mit Letzterem ist die Priorisierung von fehlerhaften Werkstücken oder die Priorisierung von ordnungsgemäß bearbeiteten Werkstücken möglich.

Der Dritte und letzte Parameter, welcher variiert werden soll, ist die Pausenplanung der Mitarbeiter. Der Referenzwert ist hierbei wie oben beschrieben eine gemeinsame 30-minütige Pause aller Mitarbeiter nach vier Stunden. Alternativ hierzu soll beobachtet werden, wie das System sich bei einer gestaffelten Pausenverteilung

verhält. Hierzu sollen die Pausen der Mitarbeiter nacheinander stattfinden. Die Pausen finden hierbei entweder von vorne nach hinten oder von hinten nach vorne im Hinblick auf die Produktionslinie statt. Der erste Mitarbeiter soll seine Pause nach zwei Stunden beginnen. Da in Arena für jeden Mitarbeiter ein eigener Pausenplan geschrieben werden muss, ist die jede Variation diese Parameters mit viel Zeitaufwand verbunden. Aus diesem Grund wird auf weitere Variationen dieses Parameters verzichtet .

Das manuelle Parametertuning soll mit Hilfe der Ein-Faktor-Methode durchgeführt werden. Hierzu soll zuerst die Pausenverteilung, dann die Priorisierung innerhalb der Warteschlangen und zum Schluss die Taktzeit variiert werden. Hierbei soll mit den oben genannten Referenzwerten begonnen werden. Wenn nach der Variation eines Parameters eine Optimierung des GP erkennbar wird, soll die optimalste Parametereinstellung als Grundlage für die nachfolgenden Parametervariationen genutzt werden.

4.3 Simulationsexperiment

Nach der Erzeugung des Simulationsmodells und der Konkretisierung des Simulationsexperiments, kann nun das Simulationsexperiment durchgeführt werden. Als Erstes soll der Parameter der Pausenplanung variiert werden. Daraus folgen die Ergebnisse aus Tabelle 4.

Taktzeit [s]	Priorisierung	Pausenplanung	Output [Stück]	ØDLZ [h]	Min. DLZ [h]	Max DLZ [h]	Ø WZ [h]
220	FIFO	Gemeinsam	100	1.2968	0.3356	2.9931	0.9348
220	FIFO	Gestaffelt M1 →P2	82	1.6259	0.3356	4.1188	1.2697
220	FIFO	Gestaffelt P2 →M1	103	1.0838	0.3356	2.3459	0.7256

Tabelle 4: Variation der Pausenplanung

Es ist zu erkennen, dass der Output und die DLZ bei einer Staffelung der Pausen von M1 zu P2 deutlich sinken. Bei einer Staffelung der Pausenzeiten von P2 zu M1 hingegen ist ein geringer Anstieg des Outputs und eine deutliche Senkung der durchschnittlichen DLZ zu erkennen. Folglich wird die Staffelung von P2 zu M1 als optimierter Parameter für die Pausenzeitordnung angenommen.

Als nächstes soll der Parameter der Prioritätengestaltung an den Warteschlangen variiert werden.

Taktzeit [s]	Priorisierung	Pausen- planung	Output [Stück]	ØDLZ [h]	Min. DLZ [h]	Max DLZ [h]	Ø WZ [h]
220	FIFO	Gestaffelt P2 →M1	103	1.0838	0.3356	2.3459	0.7256
220	LIFO	Gestaffelt P2 →M1	103	0.6332	0.3356	2.1305	0.2851
220	Fehlerhaftes WS	Gestaffelt P2 →M1	104	1.1421	0.3356	2.3532	0.7829
220	Fehlerfreies WS	Gestaffelt P2 →M1	105	0.8092	0.3356	3.3357	0.4607

Tabelle 5: Variation der Priorisierung

Beim LIFO-Prinzip bleibt der Output gleich, aber die DLZ sinkt erheblich. Es ist erkennbar, dass sich der Output bei der Priorisierung der Neuteile oder Reparaturteile minimal steigert. Die DLZ bei der Priorisierung von Neuteilen sinkt und steigt leicht bei der Priorisierung von Reparaturteilen. Wenn jedoch zusätzlich die maximale DLZ beobachtet wird, erkennt man, dass die Abweichung bei der Priorität von Neuteilen um eine Stunde höher ist als bei der Priorität der Reparaturteile. Aufgrund dieser Unsicherheit soll die Parametervariation der Taktzeit auf die beiden letzteren Priorisierungstypen angewandt werden.

In beiden Fällen ist erkennbar, dass der Output hohe Werte für geringe Taktzeiten annimmt, die DLZ jedoch bei geringen Taktzeiten signifikant zunimmt. Wenn zusätzlich zur DLZ die durchschnittliche Wartezeit (WZ) der Entitäten herangezogen wird, wird deutlich, dass diese bei geringer Taktzahl den Großteil der DLZ ausmacht. Wenn weiterführend die durchschnittlichen WZ oder die durchschnittliche Anzahl der Entitäten aufgeteilt nach den verschiedenen Puffern eingesehen wird, wird deutlich, dass geringe Taktzeiten lediglich zu einer Anstauung von Entitäten an den ersten beiden Puffern führt und zu keiner Outputsteigerung. Bei Betrachtung der einzelnen Werte, wird deutlich, dass sich die folgende Parametereinstellung als günstig herausstellt.

Taktzeit [s]	Priorisierung	Pausenplanung	Output [Stück]	ØDLZ [h]	Min. DLZ [h]	Max DLZ [h]	Ø WZ [h]
240	Fehlerhaftes WS	Gestaffelt P2 →M1	107	0.8372	0.3343	1.6854	0.4825

Tabelle 6: Optimale Parametereinstellung

Für diese Parameter Variation steigert sich der durchschnittliche Output im Vergleich zur anfänglichen Einstellung um 7 und die durchschnittliche DLZ sinkt um ca. eine halbe Stunde. Auch größere Anstauungen an den ersten beiden Pufferplätzen können durch diese Parametervariation vermieden werden.

4.4 Verifikation & Validierung

Aufgrund des geringen Umfangs des Simulationsmodells, ist davon auszugehen, dass eine abschließende V&V ausreichend ist und eine V&V während der Modellbildung keinen erheblichen Vorteil mit sich gebracht hätte. Da das Simulationsmodell auf einem Fallbeispiel beruht, welches selbst nur eine modellhafte Beschreibung der Realität ist, entfallen viele mögliche Fehlerquellen, die bei der Überführung der Realität ins Modell zu Fehlern führen könnten. Die exemplarische Durchführung des Simulationsexperiments in Kapitel 4.3 und die plausiblen Ergebnisse verifizieren und validieren das Simulationsexperiment schon teilweise.

Dennoch sollen V&V-Techniken genutzt werden, um die Glaubwürdigkeit des Modells zu steigern. Die Simulationsergebnisse aus dem Kapitel 4.3 können die Grundlage eines Kontinuitätstest bilden. Hierbei sind vor allem die Tabellen aus Anhang 4 und 5 interessant. Bei diesen wird der stetige Parameter der Taktzeit variiert. In beiden Tabellen wird deutlich, dass die Variationen der Taktzeit zu verhältnismäßigen Veränderungen und keinen Ausreißern des Outputs bzw. der DLZ führen.

Eine weitere Möglichkeit zur V&V stellt der Grenzwertest dar. Zur erleichterten Durchführbarkeit des Grenzwerttests, wird die Wahrscheinlichkeit für die fehlerhafte Bearbeitung eines Bauteils gleich null gesetzt. Auch die Dreiecksverteilungen an den Bearbeitungs- und Prüfstationen sollen verworfen werden. Stattdessen soll der Mittelwert der jeweiligen Verteilung als fester Wert festgelegt werden. Wenn nun die Taktzeit gleich 20 Sekunden gesetzt wird, ist davon auszugehen, dass der Output in einem leicht höheren Bereich als bei den

Simulationsexperimenten zuvor ist, da die Wahrscheinlichkeit für fehlerhafte Werkstücke gleich null gesetzt worden ist. Jedoch sollte die DLZ sich, aufgrund der Anstauung der Werkstücke vor den Bearbeitungs- und Prüfstationen, immens erhöhen. Wenn der Prozess für diese Taktzeit variiert wird, ist genau dies erkennbar. Eine weitere Senkung der Taktzeit ist aufgrund der Studentenversion von Arena nicht möglich.

Das gleiche kann mit einer Erhöhung der DLZ über die Simulationszeit von 8 Stunden durchgeführt werden. Hierbei sollte der Output 1 betragen, da aufgrund der hohen Taktzeit nur ein Bauteil in der Simulationszeit als Input reingegeben wird. Die DLZ dieses Werkstücks müsste sich mit 0,3314 Stunden ergeben, da Sie nur eine Addition der Bearbeitungs- und Prüfzeiten sein müsste, weil etwaige WZ vollständig entfallen würden. Auch diese Vermutung bestätigt sich. Die Ergebnisse des Grenzwerttests sind in der Tabelle 7 erkennbar.

Taktzeit	Priorisierung	Pausenplanung	Output [Stück]	ØDLZ [h]	Ø WZ [h]
9 h	FIFO	Gemeinsam	1	0.3314	0
20 s	FIFO	Gemeinsam	118	3.8	3.47

Tabelle 7 Grenzwerttest

4.5 Diskussion & Fazit

Abschließend wird versucht eine Antwort zu der Frage, wie sich Parametertuning mit der GPO verbinden lässt zu konkretisieren. Außerdem werden die Grenzen des manuellen Parametertunings diskutiert.

Grundsätzlich kann anhand der Ergebnisse aus Kapitel 4.3 davon ausgegangen werden, dass es möglich ist durch das Parametertuning GP wesentlich zu verändern. In unserem Beispiel wird es deutlich, dass es durch das manuelle Parametertuning möglich wird die DLZ zu senken und den Output zu erhöhen. Jedoch wird bei der Ausführung des Experiments deutlich, dass das manuelle Parametertuning in der GPO einige Herausforderungen mit sich bringt. Eine GPO unter Nutzung von Parametertuning hat den großen Vorteil, das in den realen GP nicht aktiv

eingegriffen werden muss. Da es sich nur um geringfügige Änderungen am Prozess handeln, ist von einer hohen Akzeptanz und Bereitschaft der Mitarbeiter auszugehen.

Schon in der Experimentplanung ist es wichtig herauszustellen, welche Parameter genutzt werden sollen, um das Parametertuning durchzuführen. In dem Simulationsexperiment dieser Arbeit, wurden hierzu die Parameter Taktzeit, Pausenplanung und Priorisierung in den Warteschlangen ausgewählt. Viele andere Parameter können gar nicht erst variiert werden. Beispielsweise ist die Bearbeitungszeit an den Bearbeitungs- und Prüfstationen nicht einstellbar. Somit entfallen viele Parameter für das Parametertuning. Daher muss vor einer Anwendung von Parametertuning im Falle einer GPO notwendigerweise geprüft werden, ob in dem GP überhaupt genügend variable Parameter existieren,

Auch bei zu vielen Parametern dieser Art, ist das manuelle Parametertuning nicht geeignet. Wenn zu viele Parametervariationen und -kombinationen existieren, ist der Arbeits- und Zeitaufwand für das manuelle Parametertuning enorm. Dies wird auch dann deutlich, wenn Parameter stetig sind. In Kapitel 4.2 wurde für das Simulationsexperiment die Taktzeit als ein stetiger Parameter für das Parametertuning gewählt. In der Ausführung des Experiments in Kapitel 4.3 wurde deutlich, dass die Taktzeit nur in einem Zeitbereich von 150 bis 300 Sekunden in 10 Sekunden Schritten variiert werden kann, da der Zeitaufwand mit jeder Parameterkombination erheblich ansteigt. Dies ist vor allem bei Parametern mit hoher Sensitivität kritisch zu betrachten. Hierbei ist es auch wichtig herauszustellen, dass es bei vielen variierbaren Parametern, zunehmend schwieriger wird Abhängigkeiten zwischen diesen zu erkennen. Hierdurch zeichnet sich ab, dass das manuelle Parametertuning nur für Simulationsmodelle geringen Umfangs, mit wenigen möglichen Parameterkombinationen sinnvoll anwendbar ist.

Die Relevanz dieser Arbeit wird deutlich, wenn Sie mit der bestehenden Literatur verglichen wird. Es ist erkennbar, dass es zwar Literatur über die GPO und das Parametertuning gibt, jedoch existieren keine Arbeiten, die beide Themen miteinander zu verbinden versuchen. Beide Themen werden aktuell nur getrennt voneinander untersucht. Diese Arbeit soll als Anstoß für weitere Untersuchungen dienen, wie beide Themen mit einander verbunden werden können und welche Vorteile dies mit

sich bringen kann. Sie soll an erster Stelle Interesse an diesem neuen Forschungsgegenstand erzeugen.

Das Anwendungsbeispiel, welches in dieser Arbeit verwendet worden ist, eignet sich aufgrund seiner Schlichtheit gut zur ersten Untersuchung der Frage, wie sich die GPO mit dem Parametertuning verbinden lässt. Dennoch ist anzumerken, dass das Anwendungsbeispiel selbst auf einer modellhaften Beschreibung von Rabe et al. (2008) beruht. Die Betrachtung eines realen Systems bzw. einer realen Produktionslinie fehlt. In weiteren Untersuchungen wäre es wichtig Simulationsergebnisse mit der Realität zu vergleichen. Beispielweise könnte das Ermüdungsverhalten von Mitarbeitern in Abhängigkeit der Pausenplanung, die Produktivität einzelner Mitarbeiter und somit den ganzen GP beeinflussen. Auch die Untersuchung mittels weiterer Simulationsexperimente und Anwendungsbeispielen, wäre hilfreich die Forschungsfrage genauer zu beantworten. Durch ein einziges Anwendungsbeispiel ist es nicht möglich alle Anwendungsbeispiele und Facetten des Parametertunings für die GPO zu untersuchen. Parametertuning könnte auch dazu genutzt werden, geplante Investitionen (z.B. eine bessere Arbeitsmaschine) zu beurteilen.

Dass aber eine Verbindung der Themen GPO und Parametertuning möglich ist und wie solch eine Verbindung aussehen kann, kann durch das Simulationsexperiment gezeigt werden. Dass Parameter, die variiert werden können, gefunden wurden und eine Variation dieser zu einer Änderung des Outputs und der DLZ führen zeigt, dass es möglich ist GP durch Parametertuning maßgeblich positiv zu beeinflussen. Somit kann die Frage, ob eine Verbindung möglich ist, eindeutig mit ja beantwortet werden. Mit dem Simulationsexperiment wird ein Beispiel gezeigt, wie sich die Themen GPO und Parametertuning miteinander verbinden lassen. Das manuelle Parametertuning mit der Ein-Faktor Methode, stellt eine Möglichkeit dar, wie die Themen GPO und Parametertuning verbunden werden können. Diese Art und Weise ist sehr einfach anzuwenden. Jedoch eignet sie sich nur für GP geringen Umfangs. Das manuelle Parametertuning mit der Ein-Faktor-Methode ist nicht die einzige Möglichkeit GPO mit Parametertuning zu betreiben. Es ist davon auszugehen, dass sie lediglich eine von Vielen Möglichkeiten darstellt. Verbunden mit der Tatsache, dass das Parametertuning eine risikoarme Möglichkeit zur GPO darstellen kann, sind weitere Untersuchungen auf diesem Gebiet empfehlenswert.

5. Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, zu untersuchen wie sich das Parametertuning mit der GPO verbinden lässt. Zu Beginn war es wichtig, die hierzu benötigten Begriffe zu definieren. Die zu verwendenden Modellierungs- und Simulationsansätze wurden ebenfalls vorgestellt. Nach der Vorstellung des zu Modellierenden GP, wurde der GP mittels EPK modelliert und anschließend in Arena implementiert. Um GPO unter Nutzung von Parametertuning durchzuführen, wurden Parameter herausgestellt, welche variiert und für das Parametertuning genutzt werden können. Nach der Erstellung des Experimentplans mit Hilfe der Ein-Faktor-Methode wurde das Simulationsexperiment durchgeführt. Es hat sich herausgestellt, dass das Parametertuning zu einer Optimierung des GP führen kann. Jedoch wurden auch die Grenzen des manuellen Parametertunings deutlich.

Aus dieser Arbeit wird erkennbar, dass es noch viel Forschungsbedarf bei der GPO unter Nutzung von Parametertuning gibt. In Kapitel 4.4 wurde angemerkt, dass das manuelle Parametertuning mit Hilfe der Ein-Faktor-Methode nur eine Möglichkeit ist, die Themen GPO und Parametertuning miteinander zu verbinden. Beispielsweise war das automatische Parametertuning kein Untersuchungsgegenstand dieser Arbeit. Da das automatische Parametertuning auch bei größeren Simulationsmodellen und bei einer Vielzahl an Parameterkombinationen benutzt werden kann, sollten Untersuchungen in diesem Bereich angestrebt werden. Eine genaue Differenzierung von Anwendungsbereichen manuellen und automatischen Parametertunings wäre daher eine sinnvolle Ergänzung zu dieser Arbeit.

Literaturverzeichnis

Becker, Jörg; Mathas, Christoph; Winkelmann, Axel (2009): Geschäftsprozessmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg.

Elizabeth Montero; María-Cristina Riff; Bertrand Neveu (2014): Applied soft Computing. A beginner's guide to tuning methods (Volume 17), S. 39–51.

Gadatsch, Andreas (2010): Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis ; eine Einführung für Studenten und Praktiker. 6., aktualisierte Aufl. Wiesbaden: Vieweg (Studium).

Grimme, Christian; Bossek, Jakob (2018): Einführung in die Optimierung. Münster: Springer.

Gutenschwager, Kai; Rabe, Markus; Spieckermann, Sven; Wenzel, Sigrid (2017): Simulation in Produktion und Logistik. Berlin, Heidelberg: Springer Berlin Heidelberg.

Huang, Changwu; Li, Yuanxiang; Yao, Xin (2020): A Survey of Automatic Parameter Tuning Methods for Metaheuristics. In: *IEEE Trans. Evol. Computat.* 24 (2), S. 201–216. DOI: 10.1109/TEVC.2019.2921598.

Kaeseler, Jens (1996): Marktanalyse Geschäftsprozeßoptimierung. Softwarewerkzeuge zur Unterstützung der Geschäftsprozeßplanung und -optimierung. Dortmund: Verl. Praxiswissen (Fabrikorganisation).

Laguna, Manuel; Marklund, Johan (2013): Business Process Modeling, Simulation and Design, Second Edition. 2nd ed. Hoboken: CRC Press. Online verfügbar unter <http://gbv.ebib.com/patron/FullRecord.aspx?p=1662986>.

Lindsay, Ann; Downs, Denise; Lunn, Ken (2003): Business processes—attempts to find a definition. In: *Information and Software Technology* 45 (15), S. 1015–1019. DOI: 10.1016/S0950-5849(03)00129-0.

Obermeier, Stefan; Fischer, Herbert; Fleischmann, Albert; Dirndorfer, Max (2014): Geschäftsprozesse realisieren. Wiesbaden: Springer Fachmedien Wiesbaden.

Palleduhn, Dirk Uwe; Neuendorf, Herbert (2013): Geschäftsprozessmanagement und integrierte Informationsverarbeitung. München: Oldenbourg Verlag.

Rockwell Automation (2021). Hg. v. Rockwell Automation. <https://www.rockwellautomation.com/en-us/products/software/arena-simulation/academic.html>.

Rosenkranz, Friedrich (2006): Geschäftsprozesse. Modell- und computergestützte Planung. 2., verb. Aufl. Berlin: Springer. Online verfügbar unter http://deposit.dnb.de/cgi-bin/dokserv?id=2671277&prov=M&dok_var=1&dok_ext=htm.

Schwarz, Lothar; Neumann, Tim; Teich, Tobias (2018): Geschäftsprozesse praxisorientiert modellieren. Berlin, Heidelberg: Springer Berlin Heidelberg.

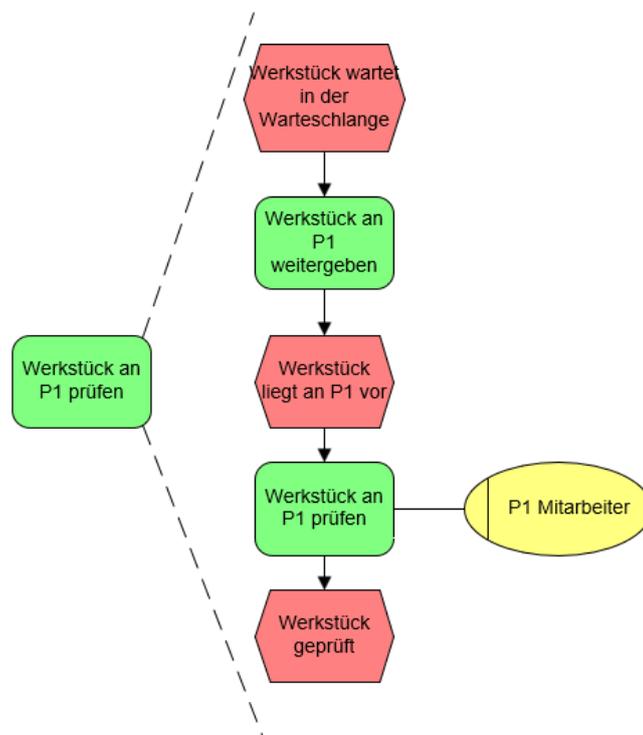
Shannon, Robert E. (1981): Test For Verification and Validation of Computer Simulation Models. 1981 Winter Simulation Conference Proceeding, S. 573–577.

Staud, Josef L. (2006): Geschäftsprozessanalyse. Ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware. 3. Aufl. Berlin: Springer. Online verfügbar unter http://deposit.dnb.de/cgi-bin/dokserv?id=2748982&prov=M&dok_var=1&dok_ext=htm.

Wicharz, Ralf (2018): Strategie: Ausrichtung von Unternehmen auf die Erfolgslogik ihrer Industrie. Wiesbaden: Springer Fachmedien Wiesbaden.

Anhang

Anhang 1: Prüfstation als Funktion.....	S.29
Anhang 2: GP als EPK.....	S.30
Anhang 3: Simulationsmodell des GP	S.31
Anhang 4: Variation der Taktzeit - Fehlerhafte WS	S.32
Anhang 5: Variation der Taktzeit – Fehlerfreie WS	S.33



Anhang 1: Prüfstation als Funktion

Parameterkombination							
Taktzeit [s]	Priorisierung	Pausenplanung	Ø Output [Stück]	Ø DLZ [h]	Min. DLZ [h]	Max. DLZ [h]	Ø WZ [h]
150	Fehlerhaftes WS	Gestaffelt P2 → M1	105	2.0452	0.3278	3.8046	1.6899
160	Fehlerhaftes WS	Gestaffelt P2 → M1	107	1.9844	0.5219	3.5092	1.6308
170	Fehlerhaftes WS	Gestaffelt P2 → M1	107	1.7211	0.3525	3.6653	1.3716
180	Fehlerhaftes WS	Gestaffelt P2 → M1	106	1.6732	0.3471	2.8064	1.3185
190	Fehlerhaftes WS	Gestaffelt P2 → M1	102	1.5137	0.3411	3.237	1.1575
200	Fehlerhaftes WS	Gestaffelt P2 → M1	105	1.3824	0.3388	2.9091	1.0252
210	Fehlerhaftes WS	Gestaffelt P2 → M1	107	1.262	0.336	2.4441	0.9042
220	Fehlerhaftes WS	Gestaffelt P2 → M1	104	1.1421	0.3356	2.3532	0.7829
230	Fehlerhaftes WS	Gestaffelt P2 → M1	106	1.0059	0.3259	1.8545	0.6455
240	Fehlerhaftes WS	Gestaffelt P2 → M1	107	0.8372	0.3343	1.6854	0.4825
250	Fehlerhaftes WS	Gestaffelt P2 → M1	100	0.8004	0.3279	1.5453	0.4406
260	Fehlerhaftes WS	Gestaffelt P2 → M1	99	0.8325	0.3294	1.7734	0.4687
270	Fehlerhaftes WS	Gestaffelt P2 → M1	99	0.6137	0.3273	1.1778	0.2601
280	Fehlerhaftes WS	Gestaffelt P2 → M1	95	0.6502	0.3249	1.359	0.2904
290	Fehlerhaftes WS	Gestaffelt P2 → M1	91	0.6514	0.3232	1.5255	0.2962
300	Fehlerhaftes WS	Gestaffelt P2 → M1	91	0.5759	0.3142	1.0622	0.2177

Anhang 4 Variation der Taktzeit - Fehlerhafte WS

Parameterkombination							
Taktzeit [s]	Priorisierung	Pausenplanung	Ø Output [stück]	Ø DLZ [h]	Min. DLZ [h]	Max. DLZ [h]	Ø WZ [h]
150	Fehlerfreies WS	Gestaffelt P2 →M1	107	1.9634	0.3278	4.3499	1.6171
160	Fehlerfreies WS	Gestaffelt P2 →M1	104	1.6314	0.359	2.76	1.289
170	Fehlerfreies WS	Gestaffelt P2 →M1	105	1.4993	0.3531	2.6605	1.1566
180	Fehlerfreies WS	Gestaffelt P2 →M1	101	1.4282	0.3504	4.6423	1.0823
190	Fehlerfreies WS	Gestaffelt P2 →M1	103	1.2219	0.3405	3.6553	0.8757
200	Fehlerfreies WS	Gestaffelt P2 →M1	108	1.2056	0.3364	3.5117	0.8601
210	Fehlerfreies WS	Gestaffelt P2 →M1	104	0.9584	0.3336	3.5462	0.6118
220	Fehlerfreies WS	Gestaffelt P2 →M1	105	0.8092	0.3356	3.3357	0.4607
230	Fehlerfreies WS	Gestaffelt P2 →M1	106	0.8192	0.3259	3.2886	0.4695
240	Fehlerfreies WS	Gestaffelt P2 →M1	99	0.8553	0.3343	2.876	0.4997
250	Fehlerfreies WS	Gestaffelt P2 →M1	101	0.6722	0.3279	1.6111	0.3236
260	Fehlerfreies WS	Gestaffelt P2 →M1	97	0.7311	0.3294	1.9178	0.373
270	Fehlerfreies WS	Gestaffelt P2 →M1	95	0.6673	0.3281	1.5023	0.3051
280	Fehlerfreies WS	Gestaffelt P2 →M1	95	0.5827	0.3249	1.2886	0.2265
290	Fehlerfreies WS	Gestaffelt P2 →M1	94	0.6121	0.328	2.1552	0.2543
300	Fehlerfreies WS	Gestaffelt P2 →M1	92	0.5462	0.3142	1.1341	0.1995

Anhang 5: Variation der Taktzeit – Fehlerfreie WS