Projektgruppe RCCARS

Realtime Controlled Cooperative Autonomous Racing System

ScrVm-Planung

Nikolai Bräuer
Michael Bukowski
Anatolij Fandrich
Tom Reske

CARL VON OSSIETZKY UNIVERSITÄT OLDENBURG
DEPARTMENT FÜR INFORMATIK

Version 2.0
10. November 2016

Veranstalter:
Prof. Dr. Werner Damm
Prof. Dr. Martin Fränzle

Betreuer:
Dipl.- Inform. Günter Ehmen
Dipl.- Inform. Stefan Puch

Inhaltsverzeichnis

1 Einführung

Das ScrVm-Planungsdokument stellt ein eigenständiges Dokument dar und beschreibt die Planung und Ausführung des *ScrVm* Prozesses (siehe Abbildung 1.1), welcher im Bericht *Anforderungsdefinition vom 09. März 2016* im Kapitelabschnitt 6.1.2 vorgestellt wurde. Dieses Dokument hat die Aufgabe den allgemeinen *ScrVm* Ablauf zu überwachen sowie eine Übersicht über den Gesamtfortschritt des Projekts zu geben. Zusätzlich unterstützt es bei der Anforderungsverwaltung. Im Weiteren bezieht sich dieses Dokument zum Teil auf Kapitelabschnitte sowie Literaturreferenzen des Berichts *Anforderungsdefinition vom 09. März 2016*. Ist dies der Fall, so wird das - wie im vorherigen Satz zu sehen - kenntlich gemacht.

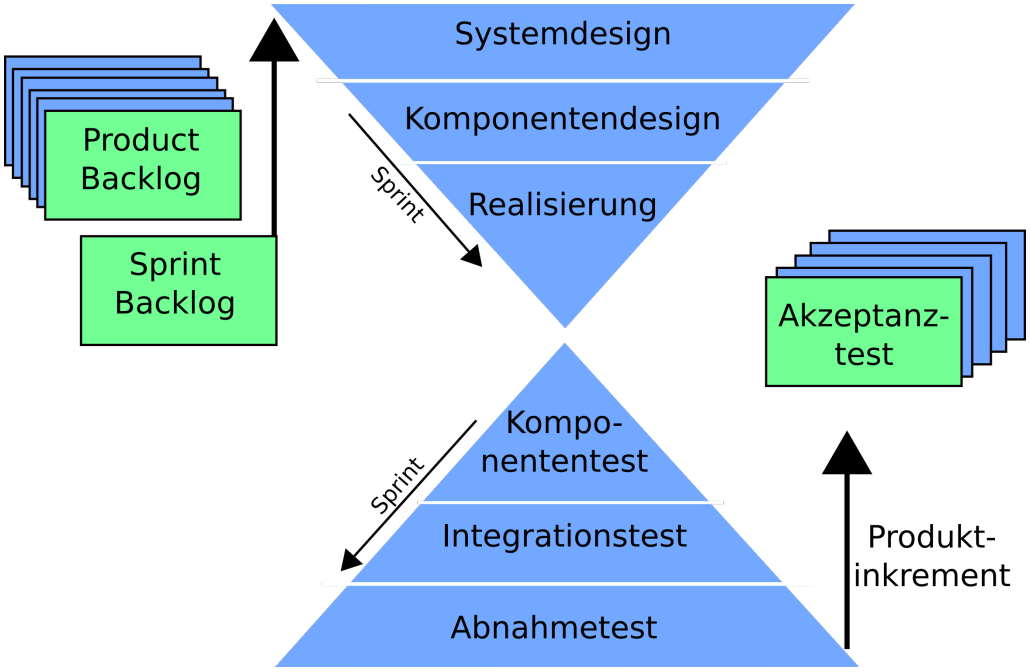

Abbildung 1.1: Beschreibung über den Ablauf des *ScrVm* Vorgehensmodells.

Im folgenden Kapitel werden die zuvor erhobenen funktionalen sowie nicht funktionalen Softwareanforderungen als Product Backlog aufgeführt. Die Rahmenbedingungen werden nicht mit aufgeführt und nicht anhand des *ScrVm* Prozess abgearbeitet, da diese Be-

dingungen darstellen, die vorliegen müssen, um das erste Szenario (siehe Kapitelabschnitt 2.4 aus dem Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllen zu können. Die funktionalen sowie nichtfunktionalen Hardwareanforderungen werden im Hardware-Evaluationskapitel des Berichts *Architektur und Schnittstellenspezifikation vom 07. Juni 2016* evaluiert sowie in der Testdatenbank aufgeführt.
Das Kapitel 3 beinhaltet mögliche zurückgeschobene Anforderungen, welche daraufhin höher priorisiert werden und so im nächsten *ScrVm* Sprint in den Sprint Backlog aufgenommen werden müssen. In Kapitel 4 werden im Anschluss die einzelnen Sprints dargelegt. Darunter sind Informationen wie das ausführende Team und der Bearbeitungszeitraum zu finden. Außerdem werden die Sprint Backlogs festgehalten. Die jeweiligen aufgestellten Testfälle werden in der Testdatenbank aufgeführt und genauer spezifiziert. Das allgemeine Testen sowie Überprüfen der Testfälle ist im Testbericht, welcher aus der Testdatenbank generiert wird, sowie im Bericht *Architektur und Schnittstellenspezifikation vom 07. Juni 2016* ausführlich beschrieben. Zudem werden nicht erfüllte, sowie neu aufgenommene Anforderungen festgehalten.
Da der Product Backlog bedingt aus Redundanzen aus dem Bericht *Anforderungsdefinition vom 09. März 2016* inkludiert wurde, ist ein leeren des Product Backlogs - so wie es im *ScrVm* Prozess üblich ist - technisch nicht möglich. Aus diesem Grund sind in Kapitel 5 alle erfüllte funktionalen sowie nicht-funktionalen Softwareanforderungen aufgeführt.

2 Product Backlog

Dieses Kapitel beschreibt den Product Backlog, welcher sich aus den Softwareanforderungen des Kapitel 4 aus dem Bericht *Anforderungsdefinition vom 09. März 2016* ableitet. Die Anforderungen sind aufgeteilt in funktionale sowie nicht funktionale Anforderungen. Dadurch bedingt, dass die Anforderungen des Product Backlogs aus dem Bericht *Anforderungsdefinition vom 09. März 2016* referenziert sind, wird sich auch durch im *ScrVm* Prozess erfüllte Anforderungen der Product Backlog optisch nicht verkleinern. Es ist deshalb auch nicht möglich, erfüllte Anforderungen zu markieren, da sich alle Änderungen auch auf das Dokument *Anforderungsdefinition vom 09. März 2016* auswirken würden. Es werden jedoch eventuell zurückgeführte Anforderungen im Kapitel 3 und alle abgearbeiteten Anforderungen im Kapitel 5 aufgeführt. Das bedeutet, dass sobald keine Anforderungen in Kapitel 3 und alle Referenzen der Anforderungen aus dem Product Backlog im Kapitel 5 zu finden sind, alle Anforderungen erfüllt wurden.

2.1 Funktionale Softwareanforderungen

SW1 **Die Software des Subsystems 2 muss fähig sein, die Pose der Fahrzeuge zu bestimmen.**

Diese Basisanforderung ergibt sich direkt aus der Systemanforderung Sys1.2. Die folgenden Unteranforderungen spezifizieren die geforderten Eigenschaften und Fähigkeiten der Positionsbestimmungssoftware, welche in Kapitelabschnitt 3.2.7 der Systembeschreibung beschrieben wurden. Auch die in Kapitelabschnitt 3.2.8 erläuterten sicherheitsrelevanten Aspekte des Subsystems 2 werden berücksichtigt.

SW1.1 Die Positionsbestimmungssoftware muss fähig sein, Bilder von der Kamera zu empfangen.

SW1.2 Die Positionsbestimmungssoftware muss fähig sein, aus den Bilddaten der Kamera die Pose des Fahrzeugs zu erkennen.

SW1.2.1 Die Positionsbestimmungssoftware muss fähig sein, die Bilder der Kamera nachzubearbeiten.

Dazu gehört beispielsweise das Entzerren und das Entrauschen des Bildes.

SW1.2.2 Die Positionsbestimmungssoftware muss durch Erkennung der Reflexionsmarker die Fahrzeuge erfassen.

Angaben zu den Reflexionsmarkern sind in Kapitelabschnitt 3.1 der Systembeschreibung zu finden.

SW1.2.3 Die Positionsbestimmungssoftware muss die zweidimensionalen Fahrzeugkoordinaten feststellen.

SW1.2.4 Die Positionsbestimmungssoftware muss die Ausrichtung der Fahrzeuge feststellen.

SW1.2.5 Die Positionsbestimmungssoftware muss die Identität der Fahrzeuge anhand der Anordnung der Reflexionsmarker feststellen.

SW1.3 Die Positionsbestimmungssoftware muss die Daten der Fahrzeugposen an die Regelungssoftware weiterleiten.

SW1.4 Die Positionsbestimmungssoftware muss fähig sein, die Bilderfassung, die Erkennung der Fahrzeugpose und die Weiterleitung der ermittelten Daten an die Regelungssoftware in höchstens 10 ms durchzuführen.

SW1.5 Positionsbestimmungssoftware muss fähig sein, die Lage der Rennstrecke zu erfassen.

Sowohl im Rahmen der Initialisierung als auch während des Rennbetriebs muss die Positionsbestimmung auch die Lage der Rennstrecke erfassen, da die Fahrzeugposen, welche an die Control-Unit übermittelt werden, auf den relativen Koordinaten der Rennstrecke basiert (siehe Kapitelabschnitt 3.4.4).

SW1.6 Die Positionsbestimmungssoftware muss Störungen erkennen.

SW1.6.1 Die Positionsbestimmungssoftware muss erkennen, ob die Rennstrecke verschoben wurde.

Dazu gehört auch das Bewegen der Kamera und das Verschieben des Gestells, da dies für das System die selben Auswirkungen in Bezug auf die Erfassung der Fahrzeuge und der Rennstrecke hat, wie das Verschieben der Rennstrecke selbst. Leichte Erschütterungen aus der Umgebung der Rennstrecke oder Ungenauigkeiten bei der Bilderfassung müssen hier jedoch toleriert werden. Als leichte Erschütterungen ist beispielsweise das Auftreten von

Personen, welche an der Rennstrecke vorbeigehen, zu verstehen. Diese Toleranzen sind festzulegen, sobald die Entwicklung der Positionsbestimmung soweit vorangeschritten ist, dass die Lage der Rennstrecke erfasst werden kann.

SW1.6.2 Falls die Positionsbestimmungssoftware eine Störung erkennt, muss sie dies der Systemsteuerungssoftware mitteilen.

Die Systemsteuerungssoftware (siehe Softwareanforderung SW3) muss in diesem Fall dafür sorgen, dass die Fahrzeuge angehalten werden und das System in den Fehlerzustand (siehe Kapitelabschnitt 3.6) versetzt wird.

SW2 **Die Software des Subsystems 3 muss fähig sein, ein Fahrzeug autonom über eine vorgegebene Rennstrecke zu steuern.**

Diese Basisanforderung ergibt sich direkt aus der Systemanforderung Sys1.3. Die folgenden Unteranforderungen definieren die geforderten Eigenschaften und Fähigkeiten der Regelungssoftware, welche in Kapitelabschnitt 3.3.4 der Systembeschreibung beschrieben wurden. Die in Kapitelabschnitt 3.3.5 erläuterten sicherheitsrelevanten Aspekte des Subsystems 3 werden ebenfalls berücksichtigt. Informationen zur Streckenführung sind Kapitelabschnitt 3.4.2 zu entnehmen.

SW2.1 Das Subsystem 3 muss eine Regelungssoftware für die autonome Steuerung des Fahrzeugs besitzen.

SW2.1.1 Die Regelungssoftware muss die Daten der Positionsbestimmung empfangen.

SW2.1.2 Die Regelungssoftware muss aus den Daten der Positionsbestimmung die Regelungsparameter für die autonome Steuerung des Fahrzeugs berechnen.

SW2.1.3 Die Regelungssoftware muss fähig sein, das Fahrzeug entlang einer vorgegeben Route zu steuern.

Wenn die Regelungssoftware gestartet wird, bekommt sie die vom Fahrzeug zu fahrende Route als Eingabeparameter übergeben. Im Rahmen der ersten Ausbaustufe des Projektes *RCCARS* soll das Fahrzeug in der Mitte der Fahrbahn fahren (siehe Kapitelabschnitt 3.4.2).

SW2.1.4 Die Regelungssoftware muss den Verlauf der Rennstrecke und die vom Fahrzeug zu fahrende Route als Eingabeparameter empfangen.

SW2.1.5 Die Regelungssoftware muss das Fahrzeug gegen den Uhrzeigersinn um die Rennstrecke steuern.

SW2.1.6 Die Regelungssoftware muss fähig sein, das Fahrzeug mit einer Durchschnittsgeschwindigkeit von mindestens 1,5 m/s über eine vorgegebene Rennstrecke zu steuern.

Die Anforderung leitet sich aus Systemanforderung Sys3 ab.

SW2.1.7 Das Fahrzeug darf die Kurvengeschwindigkeit von 1,2 m/s nicht überschreiten.

Damit das Fahrzeug sicher autonom gesteuert werden kann, muss es in der Kurve die Spur halten. Empirische Tests im Rahmen der Hardwareevaluation haben ergeben, dass die *dNaNo*-Fahrzeuge des verwendeten Typs *FX-101* fähig sind, bei einer Geschwindigkeit von 1,2 m/s auf der verwendeten Rennstrecke *Mini-Z Grand Prix Circuit 30* in der Kurve zuverlässig die Spur zu halten. Dies gilt sowohl für 90° als auch für 180° Kurven.

SW2.1.8 Die Regelungssoftware muss fähig sein, das Fahrzeug mindestens fünf Runden lang über die vorgegebene Rennstrecke zu steuern.

SW2.1.9 Die Regelungssoftware muss die Regelungsdaten für die Fahrzeugsteuerung an die CarControl übermitteln.

SW2.1.10 Die Regelungssoftware muss Unfälle und Störungen erkennen.

SW2.1.10.1 Die Regelungssoftware muss erkennen, ob das Fahrzeug den befahrbaren Bereich der Rennstrecke verlassen hat.

Der befahrbare Bereich der Rennstrecke wird Kapitelabschnitt 3.4.4 der Systembeschreibung definiert.

SW2.1.10.2 Die Regelungssoftware muss fähig sein, eine Kollision des Fahrzeugs mit der Streckenbegrenzung zu erkennen.

SW2.1.10.3 Die Regelungssoftware muss erkennen, ob das Fahrzeug in die richtige Richtung fährt.

Es ist vorgesehen, dass das Fahrzeug gegen den Uhrzeigersinn gesteuert wird. Sollte dies nicht der Fall sein, muss die Regelungssoftware dies feststellen (siehe Kapitelabschnitt 3.4.2 der Systembeschreibung).

SW2.1.10.4 Die Regelungssoftware muss den Verlust des Sichtkontakts zum Fahrzeug erkennen.

Dies tritt ein, wenn sich das Fahrzeug überschlägt, die Bilderfassung gestört wird oder das Fahrzeug das Sichtfeld der Kamera verlässt. In diesen Fall gibt die Positionsbestimmungssoftware statt der Fahrzeugpose Fehlerdaten aus, so dass die Regelungssoftware erkennen kann, dass das Fahrzeug nicht erfasst werden konnte.

SW2.1.10.5 Falls keine Daten von der Positionsbestimmung empfangen werden, muss die Regelungssoftware das Fahrzeug stoppen.

Sobald das Fahrzeug für 20 ms nicht erkannt wird, muss die Regelungssoftware das Fahrzeug stoppen. Diese 20 ms entsprechen der Aufnahmezeit von zwei Kamerabildern. Ausgehend von der Durchschnittsgeschwindigkeit von 1,5 m/s (siehe Kapitelabschnitt 2.4) kann das Fahrzeug in dieser Zeit 3 cm zurücklegen. Bei dieser Geschwindigkeit muss die Regelungssoftware, fähig sein das Fahrzeug anzuhalten bevor es die Fahrbahnbegrenzung berührt. Da diese Geschwindigkeit – beispielsweise auf den langen Geraden der Rennstrecke – auch überschritten werden kann, ist es möglich, dass sich das Fahrzeug der Fahrbahnbegrenzung so schnell nähert, dass eine Kollision mit dieser unter Umständen nicht mehr vermieden werden kann. Im diesem Fall muss die Regelungssoftware das Fahrzeug sofort abbremsen, so dass die Geschwindigkeit des Fahrzeugs bei der Kollision mit der Fahrbahnbegrenzung möglichst gering ist.

SW2.1.10.6 Die Regelungssoftware muss fähig sein, den Verlust der Verbindung zur CarControl zu erkennen.

Die Regelungssoftware sendet Befehle über die CarControl an das Fahrzeug (siehe Kapitelabschnitt 3.3.3). Ist diese Verbindung unterbrochen, so ist eine Kontrolle des Fahrzeugs nicht mehr möglich. Die Kommunikation zwischen der Regelungssoftware und der CarControl wird über eine virtuelle serielle Schnittstelle hergestellt. Bei einem Verbindungsverlust wird diese automatisch geschlossen. Das Senden eines Befehls an die CarControl ist demnach nicht mehr möglich und ein Fehler wird erzeugt.

SW2.1.10.7 Die Regelungssoftware muss bei der Auswertung der Fahrzeugpose eine Plausibilitätsüberprüfung durchführen.

Siehe Kapitelabschnitt 3.3.4.

SW2.1.10.8 Falls die Regelungssoftware eine Störung erkennt, muss sie das Fahrzeug stoppen.

SW2.1.10.9 Falls die Regelungssoftware eine Störung erkennt, muss sie dies der Systemsteuerungssoftware mitteilen.

Die Systemsteuerungssoftware (siehe Softwareanforderung SW3) muss in diesem Fall nur noch dafür sorgen, dass das System in den Fehlerzustand (siehe Kapitelabschnitt 3.6) versetzt wird, da bereits die Regelungssoftware sicherstellt, dass das Fahrzeug angehalten wird (siehe Softwareanforderung SW2.1.10.8).

SW2.1.11 Bei der Entwicklung der Regelungssoftware muss das Werkzeug *SCADE* [SCA] zum Einsatz kommen.

SW2.2 Das Subsystem 3 muss eine Software für die Umwandlung der digitalen Regelungsdaten in analoge Spannungen besitzen.

SW2.2.1 Die CarControlsoftware muss die Regelungsdaten empfangen können.

SW2.2.2 Die CarControlsoftware muss die digitalen Regelungsdaten in Spannungen im Bereich zwischen 0 und 3 V umwandeln.

SW2.2.3 Die CarControlsoftware muss die in Spannungen übersetzen Regelungsdaten an die Fernsteuerung übermitteln.

SW2.2.4 Falls keine Daten von der Regelungssoftware empfangen werden, muss die CarControlsoftware das Fahrzeug stoppen.

Siehe auch Softwareanforderung SW2.1.10.6.

SW2.3 Die Software der Control-Unit (Regelungssoftware und CarControlsoftware) muss fähig sein, die Berechnung und das Weiterleiten der Steuersignale in weniger als 10 ms durchzuführen.

Die Berechnung und das Weiterleiten darf nicht länger als 10 ms dauern, da nach Ablauf dieser Zeitspanne bereits neue Daten von der Positionsbestimmung vorliegen. Siehe hierzu auch Kapitelabschnitt *Communication between computer and transmitter* in [EGH+15].

SW3 **Die Systemsteuerungssoftware muss Funktionalitäten zur Bedienung und Überwachung des Systems bieten.**

Diese Basisanforderung ergibt sich aus der Systemanforderung Sys2.
Die folgenden Unteranforderungen definieren die geforderten Eigenschaf-

ten und Fähigkeiten der Systemsteuerungssoftware, welche in Kapitelabschnitt 3.5 der Systembeschreibung beschrieben wurden. Auch der Kapitelabschnitt 3.6 zum Kontrollfluss des Systembetriebs wird an dieser Stelle berücksichtigt.

SW3.1 Die Systemsteuerungssoftware muss dem Administrator die Möglichkeit bieten, das System über ein Benutzerinterface zu bedienen.

SW3.1.1 Das Benutzerinterface muss dem Administrator die Möglichkeit bieten, zu bestätigen, dass die Inspektion erfolgreich durchgeführt wurde.

Die Anforderungen an die Inspektion werden Abschnitt 4.1 erläutert.

SW3.1.2 Das Benutzerinterface muss dem Administrator die Möglichkeit bieten, die Initialisierung des Systems starten.

Diese Anforderung leitet sich aus Systemanforderung Sys2.2 ab.

SW3.1.3 Das Benutzerinterface muss dem Administrator die Möglichkeit bieten, nach der Initialisierung den Rennbetrieb zu starten.

Diese Anforderung leitet sich aus Systemanforderung Sys2.4 ab.

SW3.1.4 Das Benutzerinterface muss dem Administrator jederzeit die Möglichkeit bieten, den Rennbetrieb zu stoppen.

Diese Anforderung leitet sich aus Systemanforderung Sys2.5 ab.

SW3.1.5 Das Benutzerinterface muss dem Administrator jederzeit die Möglichkeit bieten, das System auszuschalten.

Diese Anforderung leitet sich aus Systemanforderung Sys2.3 ab.

SW3.1.6 Das Benutzerinterface muss dem Administrator jederzeit die Möglichkeit bieten, das System manuell in den Fehlerzustand zu versetzen.

Diese Anforderung leitet sich aus Systemanforderung Sys2.6 ab.

SW3.1.7 Das Benutzerinterface muss den Administrator über den aktuellen Systemzustand informieren.

SW3.2 Falls ein sicherheitskritischer Fehler auftritt, muss die Systemsteuerungssoftware das System in einen sicheren Fehlerzustand überführen.

SW3.2.1 Die Systemsteuerungssoftware muss die Fehlermeldungen, der Positionsbestimmungssoftware und der Regelungssoftware empfangen.

Siehe Softwareanforderungen SW1.6.2 und SW2.1.10.9.

SW3.2.2 Falls ein sicherheitskritischer Fehler auftritt, muss die Systemsteuerungssoftware allen Instanzen der Regelungssoftware mitteilen, dass die Fahrzeuge sofort gestoppt werden müssen.

SW4 Die Systemsteuerungssoftware, die Positionsbestimmungssoftware und alle Instanzen der Regelungssoftware müssen parallel ausgeführt werden können.

Die Positionsbestimmungssoftware und die Regelungssoftware müssen im Rahmen der ersten Ausbaustufe des Projektes *RCCARS* auf dem selben Rechner ausgeführt werden. Um die Leistung des Systems zu erhöhen, müssen diese parallel ausgeführt werden. So kann bereits ein neues Kamerabild verarbeitet werden, während noch Regelungsdaten berechnet werden, welche auf dem vorherigen Kamerabild basieren.

SW5 Die Systemsteuerungssoftware, die Positionsbestimmungssoftware und die Regelungssoftware müssen mit der höchsten Priorität ausgeführt werden.

Damit die Systemsteuerungssoftware, Positionsbestimmungssoftware und die Regelungssoftware im Vergleich zu anderen Prozessen die höchste Priorität erhalten, muss ein Realzeitbetriebssystem als Ausführungsumgebung verwendet werden (siehe Kapitelabschnitt 3.2.8). Die Systemsteuerungssoftware muss dafür Sorgen, dass die genannten Softwarekomponenten die entsprechende Priorität erhalten. Daraus leitet sich die folgende Anforderung an das Betriebssystem ab.

SW5.1 Das auf dem Rechner installierte Betriebssystem muss realzeitfähig sein.

Falls im Rahmen einer späterer Ausbaustufe eine oder mehrere Control-Units auf einen oder mehreren anderen Rechnern betrieben werden als die Positionsbestimmung und die Systemsteuerungssoftware, gilt diese Anforderung auch für jeden dieser Rechner.

2.2 Nicht funktionale Softwareanforderungen

NSW1 **Die Softwarekomponenten des Systems müssen fähig sein, Information über den Systembetrieb in Textform zu protokollieren.**

Damit die Vorgänge während des Systembetriebs nachvollzogen werden können und um eine Erkennung und Auswertung von Fehlern zu ermöglichen, müssen die von den einzelnen Softwarekomponenten erzeugten Ausgaben gespeichert werden. Zu Optimierungszwecken ist es außerdem erforderlich zu erfahren, wie lange die Softwarekomponenten be-

nötigen, um ihre Aufgaben auszuführen. Da diese Protokollfunktionen die Programmlaufzeit verlängern, sollen sie optional im Rahmen eines Debugging-Modus aktiviert werden können. Die Ausgabe von Fehlerberichten ist hingegen immer erforderlich. Daraus ergeben sich die folgenden Anforderungen an die jeweiligen Softwarekomponenten.

NSW1.1 Die Positionsbestimmungssoftware muss im Debugging-Modus die an die Regelungssoftware übergebenen Parameter protokollieren.

NSW1.2 Die Positionsbestimmungssoftware muss im Debugging-Modus die Laufzeit der Bildverarbeitung protokollieren.

NSW1.3 Die Regelungssoftware muss im Debugging-Modus die berechneten Regelungssignale protokollieren.

NSW1.4 Die Regelungssoftware muss im Debugging-Modus die Laufzeit der Regelungssignalberechnung protokollieren.

NSW1.5 Die Software auf der CarControl muss im Debugging-Modus die an die Fernsteuerung übergebenen Spannungsstärken protokollieren.

NSW1.6 Die Software auf der CarControl muss im Debugging-Modus die Laufzeit der Signalumwandlung protokollieren.

NSW1.7 Falls ein Fehler auftritt müssen die betroffenen Softwarekomponenten des Systems einen Fehlerbericht erstellen.

3 Höher priorisierte Anforderungen

3.1 Anforderung SW2.1.4

- **Referenz**: SW2.1.4
- **Subsystem**: Control-Unit
- **Grund:** Neu aufgenommene Anforderung in Sprint 1.2 (siehe Kapitelabschnitt 4.2). Diese Anforderung ist nötig um den Verlauf der Rennstrecke sowie der Trajektorie einlesen zu können.
- **Auflage:** Diese Anforderung muss in Sprint 2.2 (siehe Kapitelabschnitt 4.4) in den Sprint Backlog aufgenommen werden.
- **Status:** Erfüllt in Sprint 2.2 (siehe Kapitelabschnitt 4.4).

3.2 Anforderung SW1.2.4

- **Referenz**: SW1.2.4
- **Subsystem**: Positionsbestimmung
- **Grund:** Rückgeführte Anforderung in Sprint 1.1 (siehe Kapitelabschnitt 4.1) bedingt durch Probleme in der Realisierung.
- **Auflage:** Diese Anforderung muss in Sprint 2.1 (siehe Kapitelabschnitt 4.3) in den Sprint Backlog aufgenommen werden.
- **Status:** Erfüllt in Sprint 2.1 (siehe Kapitelabschnitt 4.3).

3.3 Anforderung SW1.2.5

Erste Rückführung

- **Referenz**: SW1.2.5

- **Subsystem**: Positionsbestimmung
- **Grund:** Rückgeführte Anforderung in Sprint 1.1 (siehe Kapitelabschnitt 4.1) bedingt durch Zeitmangel.
- **Auflage:** Diese Anforderung muss in Sprint 2.1 (siehe Kapitelabschnitt 4.3) in den Sprint Backlog aufgenommen werden.
- **Status:** In Bearbeitung in Sprint 2.1 (siehe Kapitelabschnitt 4.3).

Zweite Rückführung

- **Referenz**: SW1.2.5
- **Subsystem**: Positionsbestimmung
- **Grund:** Rückgeführte Anforderung in Sprint 2.1 (siehe Kapitelabschnitt 4.5) bedingt durch Personalausfall und Entwicklungsproblemen.
- **Auflage:** Diese Anforderung muss in Sprint 3.1 (siehe Kapitelabschnitt 4.5) in den Sprint Backlog aufgenommen werden.
- **Status:** Erfüllt in Sprint 3.1 (siehe Kapitelabschnitt 4.5).

3.4 Anforderung SW2.1.2

- **Referenz**: SW2.1.2
- **Subsystem**: Control-Unit
- **Grund:** Rückgeführte Anforderung in Sprint 3.3 (siehe Kapitelabschnitt 4.7). Diese Anforderung wurde zurückgeführt, da die Regelung zwar in ersten Zügen umgesetzt wurde, diese jedoch noch verbessert werden muss.
- **Auflage:** Diese Anforderung muss in Sprint 4.1 (siehe Kapitelabschnitt 4.8) in den Sprint Backlog aufgenommen werden.
- **Status:** Erfüllt in Sprint 4.1 (siehe Kapitelabschnitt 4.8).

3.5 Anforderung SW2.1.3

- **Referenz**: SW2.1.3
- **Subsystem**: Control-Unit
- **Grund:** Rückgeführte Anforderung in Sprint 3.3 (siehe Kapitelabschnitt 4.7). Diese Anforderung wurde zurückgeführt, da die Regelung zwar in ersten Zügen umgesetzt wurde, diese jedoch noch verbessert werden muss.
- **Auflage:** Diese Anforderung muss in Sprint 4.1 (siehe Kapitelabschnitt 4.8) in den Sprint Backlog aufgenommen werden.
- **Status:** Erfüllt in Sprint 4.1 (siehe Kapitelabschnitt 4.8).

3.6 Anforderung SW2.1.6

- **Referenz**: SW2.1.6
- **Subsystem**: Control-Unit
- **Grund:** Nicht erfüllte Anforderung in Sprint 4.1 (siehe Kapitelabschnitt 4.8). Diese Anforderung wurde zurückgeführt, da die Mindestdurchschnittsgeschwindigkeit nicht erreicht wurde.
- **Auflage**: Abänderung der Mindestdurchschnittsgeschwindigkeit in Absprache mit den Auftraggebern von 1,5 auf 1,2 $\frac{m}{s}$
- **Status:** Erfüllt in Sprint 4.1 durch die Abänderung der Anforderung.

3.7 Anforderung SW5

- **Referenz**: SW5
- **Subsystem**: Positionsbestimmung, Control-Unit und Systemsteuerungssoftware
- **Grund:** Nicht erfüllte Anforderung in Sprint 3.1, 3.2 und 4.1 (siehe Kapitelabschnitt 4.5, 4.5 und 4.8). Diese Anforderung wurde zurückgeführt, da die Softwarekomponenten nicht priorisiert werden. Bedingt durch zeitliche Engpässe konnten keine Softwarepriorisierungen mehr implementiert werden. Das verwendete Betriebssystem ist jedoch realzeitfähig.

- **Auflage**: Da die Ausführung dieser Anforderung in den letzten Sprints der Projektgruppe fehlgeschlagen ist, kann keine Erfüllungsauflage gesetzt werden.
- **Status:** Nicht erfüllt.

4 Sprints

4.1 Sprint 1.1

4.1.1 Ausführendes Team

- **Team:** Positionsbestimmung
- **Teammitglied 1:** Anatolij Fandrich
- **Teammitglied 2:** Nikolai Bräuer

4.1.2 Zeitraum

- **Starttermin:** 17.03.2016
- **Endtermin:** 05.05.2016
- **Dauer:** 49 Tage

4.1.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 2 (siehe SW1), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- SW1.1
- SW1.2

4.1.4 Nicht erfüllte Anforderungen

- SW1.2.4
- SW1.2.5

4.1.5 Neu aufgenommene Anforderungen

Keine

4.2 Sprint 1.2

4.2.1 Ausführendes Team

- **Team:** Control-Unit
- **Teammitglied 1:** Michael Bukowski
- **Teammitglied 2:** Tom Reske

4.2.2 Zeitraum

- **Starttermin:** 29.03.2016
- **Endtermin:** 12.05.2016
- **Dauer:** 51 Tage

4.2.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 3 (siehe SW2), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- SW2.1.1
- SW2.2

4.2.4 Nicht erfüllte Anforderungen

Keine

4.2.5 Neu aufgenommene Anforderungen

- SW2.1.4

4.3 Sprint 2.1

4.3.1 Ausführendes Team

- **Team:** Positionsbestimmung
- **Teammitglied 1:** Anatolij Fandrich
- **Teammitglied 2:** Nikolai Bräuer

4.3.2 Zeitraum

- **Starttermin:** 06.05.2016
- **Endtermin:** 24.06.2016
- **Dauer:** 49 Tage

4.3.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 2 (siehe SW1), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- **SW1.2.4** (höher priorisiert)
- **SW1.2.5** (höher priorisiert)
- SW1.5
- SW1.6.1

4.3.4 Nicht erfüllte Anforderungen

- SW1.2.5

4.3.5 Neu aufgenommene Anforderungen

Keine

4.4 Sprint 2.2

4.4.1 Ausführendes Team

- **Team:** Control-Unit
- **Teammitglied 1:** Michael Bukowski
- **Teammitglied 2:** Tom Reske

4.4.2 Zeitraum

- **Starttermin:** 13.05.2016
- **Endtermin:** 24.06.2016
- **Dauer:** 49 Tage

4.4.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 3 (siehe SW2), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- **SW1.3** (höher priorisiert)
- **SW2.1.4** (höher priorisiert)
- SW2.1.9
- SW2.1.10.1
- SW2.1.10.2
- SW2.1.10.5
- SW2.1.10.6
- SW2.1.10.8
- SW2.1.11

4.4.4 Nicht erfüllte Anforderungen

Keine

4.4.5 Neu aufgenommene Anforderungen

Keine

4.5 Sprint 3.1

4.5.1 Ausführendes Team

- **Team:** Positionsbestimmung
- **Teammitglied 1:** Anatolij Fandrich

4.5.2 Zeitraum

- **Starttermin:** 11.07.2016
- **Endtermin:** 02.09.2016 → 07.10.2016
- **Dauer:** 56 Tage → 88 Tage

4.5.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 2 (siehe SW1), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- **SW1.2.5** (höher priorisiert)
- SW1.4
- SW1.6.2
- SW4
- SW5
- NSW1.1
- NSW1.2
- NSW1.7

4.5.4 Nicht erfüllte Anforderungen

- SW5

4.5.5 Neu aufgenommene Anforderungen

Keine

4.6 Sprint 3.2

4.6.1 Ausführendes Team

- **Team:** Systemsteuerungssoftware
- **Teammitglied 1:** Nikolai Bräuer

4.6.2 Zeitraum

- **Starttermin:** 11.07.2016
- **Endtermin:** 07.09.2016
- **Dauer:** 61 Tage

4.6.3 Sprint Backlog

Die folgenden Softwareanforderungen an die Systemsteuerungssoftware (siehe SW3), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- SW3
- SW4
- SW5
- NSW1.7

4.6.4 Nicht erfüllte Anforderungen

- SW5

4.6.5 Neu aufgenommene Anforderungen

- SW3.1.5

4.7 Sprint 3.3

4.7.1 Ausführendes Team

- **Team:** Control-Unit
- **Teammitglied 1:** Michael Bukowski (geteilte Ressource)
- **Teammitglied 2:** Tom Reske

4.7.2 Zeitraum

- **Starttermin:** 11.07.2016
- **Endtermin:** 02.09.2016
- **Dauer:** 56 Tage

4.7.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 3 (siehe SW2), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- SW2.1.2
- SW2.1.3
- SW2.1.5
- SW2.1.10.3
- SW2.1.10.4

4.7.4 Nicht erfüllte Anforderungen

- SW2.1.2
- SW2.1.3

4.7.5 Neu aufgenommene Anforderungen

Keine

4.8 Sprint 4.1

4.8.1 Ausführendes Team

- **Team:** Control-Unit
- **Teammitglied 1:** Michael Bukowski
- **Teammitglied 2:** Nikolai Bräuer
- **Teammitglied 3:** Tom Reske

4.8.2 Zeitraum

- **Starttermin:** 02.09.2016
- **Endtermin:** 07.10.2016
- **Dauer:** 35 Tage

4.8.3 Sprint Backlog

Die folgenden Softwareanforderungen des Subsystems 3 (siehe SW2), sowie alle dazugehörigen tiefer geschachtelten Anforderungen müssen anhand des *ScrVm* Vorgehensmodells (siehe Kapitelabschnitt 6.1.2 im Bericht *Anforderungsdefinition vom 09. März 2016*) erfüllt werden.

- SW2.3
- SW2.1.10
- SW2.1.3
- SW2.1.7
- SW2.1.8
- SW2.1.10.7
- SW2.1.10.9
- SW4
- SW5
- NSW1.3

- NSW1.4
- NSW1.5
- NSW1.6
- NSW1.7

4.8.4 Nicht erfüllte Anforderungen

- SW2.1.6
- SW5

4.8.5 Neu aufgenommene Anforderungen

Keine

5 Erfüllte Anforderungen

- SW1.1: Erfüllt durch Sprint 1.1 (siehe Abschnitt 4.1).
- SW1.2.1: Erfüllt durch Sprint 1.1 (siehe Abschnitt 4.1).
- SW1.2.2: Erfüllt durch Sprint 1.1 (siehe Abschnitt 4.1).
- SW1.2.3: Erfüllt durch Sprint 1.1 (siehe Abschnitt 4.1).
- SW1.2.4: Erfüllt durch Sprint 2.1 (siehe Abschnitt 4.3).
- SW1.2.5: Erfüllt durch Sprint 2.1 (siehe Abschnitt 4.3)
- SW1.3: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW1.4: Erfüllt durch Sprint 3.1 (siehe Abschnitt 4.5).
- SW1.5: Erfüllt durch Sprint 2.1 (siehe Abschnitt 4.3)
- SW1.6.1: Erfüllt durch Sprint 2.1 (siehe Abschnitt 4.3)
- SW1.6.2: Erfüllt durch Sprint 3.1 (siehe Abschnitt 4.5).
- SW2.1.1: Erfüllt durch Sprint 1.2 (siehe Abschnitt 4.2).
- SW2.1.2: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.3: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.4: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.5: Erfüllt durch Sprint 3.3 (siehe Abschnitt 4.7).
- SW2.1.7: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.8: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.9: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.10.1: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).

- SW2.1.10.2: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.10.3: Erfüllt durch Sprint 3.3 (siehe Abschnitt 4.7).
- SW2.1.10.4: Erfüllt durch Sprint 3.3 (siehe Abschnitt 4.7).
- SW2.1.10.5: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.10.6: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.10.7: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.10.8: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.1.10.9: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW2.1.11: Erfüllt durch Sprint 2.2 (siehe Abschnitt 4.4).
- SW2.2: Erfüllt durch Sprint 1.2 (siehe Abschnitt 4.2).
- SW2.3: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- SW3: Erfüllt durch Sprint 3.2 (siehe Abschnitt 4.6).
- SW4: Erfüllt durch Sprint 3.1, 3.2 sowie 4.1 (siehe Abschnitt 4.5, 4.6 und 4.8).
- NSW1.1: Erfüllt durch Sprint 3.1 (siehe Abschnitt 4.5).
- NSW1.2: Erfüllt durch Sprint 3.1 (siehe Abschnitt 4.5).
- NSW1.3: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- NSW1.4 : Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- NSW1.5: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- NSW1.6: Erfüllt durch Sprint 4.1 (siehe Abschnitt 4.8).
- NSW1.7: Erfüllt durch Sprint 3.1, 3.2 und 4.1 (siehe Abschnitt 4.5, 4.6 und 4.8).

Literaturverzeichnis

[EGH+15] EK, Johan ; GUNNARSSON, Johannes ; HELLAEUS, Viktor ; INSGARD, Viktor ; KUOSKU, Mattisa ; NORRBLOM, William: Chalmers University of Technology Goethenburg - Advanced vehicle control systems. (2015)

[SCA] Esterel Technologies - SCADE Suite. http://www.esterel-technologies.com/products/scade-suite/. – Letzter Zugriff: 12.02.2016