Sicherheitsrelevante Software in der Kerntechnik - Sicht des Gutachters

Günter Glöe

Software & Elektronik Zertifizierungsstelle (SEECERT), TÜV Nord e.V.
22525 Hamburg
GGloee@TUEV-Nord.de

Abstract: Software wird in der Kerntechnik seit etwa 30 Jahren für sicherheitsrelevante Aufgaben eingesetzt. Der Zusammenhang von Sicherheit mit anderen Qualitätsmerkmalen von Software wird dargestellt. Die Rolle des Gutachters im Zusammenhang mit sicherheitsrelevanter Leittechnik und Software wird beschrieben. Besonders eingegangen wird auf die üblicherweise unvollständigen Qualitäts- und Sicherheitsvorgaben für die Software-Entwicklung, auf vorbenutzte Software, auf Statische Analysen, auf Mängel im Code und auf den Umgang mit Standards in Entwicklung und Begutachtung. Einige bei der Begutachtung verwendete Hilfsmittel (Tools) werden genannt.

1. Von den ersten Anwendungen Anfang der 70er Jahre zum heutigen Stand

Anfang der siebziger Jahre wurde in Deutschland das erste Mal versucht, Rechnersysteme in sicherheitsrelevante Anwendungen von Kernkraftwerken zu integrieren. Das geschah etwa zeitgleich mit solchen Arbeiten z. B. in Kanada. Dabei ging aber der deutsche Ansatz, das komplette Reaktorschutzsystem auf Prozessrechnerbasis zu realisieren, über die seinerzeitigen Projekte im Ausland hinaus.

Erst in den neunziger Jahren hat sich dann die Anwendung von Rechnersystemen auf Aufgaben aller Sicherheitsstufen ausgeweitet. Dazu gehören Reaktorschutzsysteme und die vorgelagerten Begrenzungen ebenso wie Steuerungen für Hebezeuge oder Notstromdiesel.

Heute muss – bedingt durch das Alter kerntechnischer Anlagen – die installierte Leittechnik in nennenswertem Umfang gegen neue Komponenten ersetzt werden, um den bisherigen Sicherheitsstand zu erhalten. Als neue Komponenten stehen dabei vielfach nur noch rechnerbasierte Komponenten zur Verfügung. Daneben bietet die immer weiter steigende Leistungsfähigkeit rechnergestützter Leittechnik weitere Möglichkeiten, unerwünschte Betriebszustände zu detektieren und zu beherrschen. Entsprechend haben die im Kernkraftwerksbereich verbliebenen größeren Leittechnik-Anbieter durchweg rechnerbasierte Leittechnik für sicherheitsrelevante Aufgaben im Angebot.

Der Software Einsatz für sicherheitsrelevante Anwendungen ist in der Kerntechnik – national wie international – zur Selbstverständlichkeit geworden.

2. Sicherheit und andere Qualitätsmerkmale von Software

Qualität von Software ist ein Gebiet mit detaillierter Struktur. So wie bei anderen Produkten auch, wird die Qualität von Software nach Qualitätsmerkmalen strukturiert. Allgemein akzeptierte Merkmale für Software werden z.B. in DIN 66272 angegeben.

Qualitätsmerkmal	Qualitäts-Teilmerkmal
Funktionalität	Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit, IT-Sicherheit
Zuverlässigkeit	Reife, Fehlertoleranz, Wiederherstellbarkeit
Benutzbarkeit	Verständlichkeit, Erlernbarkeit, Bedienbarkeit
Effizienz	Zeitverhalten, Verbrauchsverhalten
Änderbarkeit	Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
Übertragbarkeit	Anpaßbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

Bild 1: Qualitätsmerkmale von Software

Aufbauend auf diesen Merkmalen und Teilmerkmalen lässt sich die für eine Software anzustrebende Qualität systematisch vorgeben. Dabei sollte berücksichtigt werden, dass die Merkmale untereinander Unverträglichkeiten aufweisen können. So wie hohe Motorleistung und geringer Treibstoffverbrauch nur schwer miteinander vereinbar sind, sind es auch Zuverlässigkeit und Speicherplatz-Effizienz. Auch im Software-Bereich sind herausragende Anforderungen an einzelne Merkmale nur mit herausragendem Aufwand zu erreichen.

Insbesondere zu den Merkmalen Zuverlässigkeit und Sicherheit sind die einschlägigen Vorgaben durchweg in detaillierter Weise in den einschlägigen Normen enthalten, z.B. in der IEC 61508. Die Vorgabe der mit der Software zu erreichenden Zuverlässigkeit und Sicherheit - und natürlich auch der zu erbringenden Funktionalität - ist Voraussetzung für eine erfolgreiche Entwicklung.

Das Prinzip der unklaren Ziele :
Projekte ohne klare Ziele werden ihre Ziele nie klar erreichen! (aus [VDI93])

Die vorgegebene Qualität (einschließlich der Funktionalität) kann in einem definierten Ablauf erarbeitet werden. Deshalb spricht man vom Software-**Engineering** !

Von dem Erreichen der Qualität können sich Hersteller, Kunde oder unabhängige Dritte zeitlich und aufwandsmäßig vorausplanbar ein nachvollziehbares Bild machen. Das erfolgt in der Validation, Verifikation, Prüfung oder Zertifizierung der Software. Darauf wird im nächsten Abschnitt weiter eingegangen.

Viele Software-Fehler haben ihre Ursache in frühen Entwicklungsphasen. Die Fehlerbeseitigung wird mit fortschreitender Entwicklung immer aufwendiger. Deshalb ist es vernünftig, sich bereits in den frühen Entwicklungsphasen vom Erreichen der vorgegebenen Qualität zu vergewissern.

3. Die Rolle des Gutachters

Wie der Bau von Kernkraftwerken insgesamt bedürfen auch die Errichtung und der Austausch von Leittechnik in Kernkraftwerken durchweg der Genehmigung. Voraussetzung für das Erteilen einer Genehmigung ist die Feststellung, dass die nach dem Stand von Wissenschaft und Technik erforderliche Vorsorge gegen Schäden getroffen ist. Diese Feststellung wird durchweg von Gutachtern getroffen, die deswegen häufig an Leittechnik- und Software-Projekten für Kernkraftwerke beteiligt sind.

Gutachter: Person, die bestimmt wurde, die Begutachtung durchzuführen.
Begutachtung: Analyseprozeß

- zur Feststellung, ob die Entwurfsinstanz und der Validierer ein Produkt zustande gebracht haben, das die spezifizierten Anforderungen erfüllt, und um
- zu beurteilen, ob das Produkt für seinen gedachten Anwendungszweck geeignet ist.

(aus DIN V ENV 50129; Juli 1999)

Entgegen der in manchen Projekten beobachteten Erwartung ist die Begutachtung kein Ersatz für eine ordentliche, die Qualitätsziele berücksichtigende Entwicklung. Die Begutachtung oder Überprüfung stellen das erreichte – ggf. unzulängliche Qualitätsniveau – der Software fest. Begutachtung verbessert Software nicht, verschlechtert sie auch nicht. Auch ersetzt die Begutachtung nicht Verifikation oder Validation beim Hersteller und Betreiber.

Durch die Begutachtung soll das Vertrauen in die Qualität (oder Zuverlässigkeit oder Sicherheit) begründet werden. Wo mit Software Risiko für Leben, Umwelt oder Vermögen verbunden ist, kann eine positiv ausgefallene Begutachtung (oder Überprüfung oder Zertifizierung) helfen, sich ggf. nicht dem Vorwurf der groben Fahrlässigkeit auszusetzen.

Abhängig vom Risiko, das mit einer zu begutachtenden Software verbunden ist, werden die verwendeten Analysemethoden und der Umfang der analysierten Software-Teile (das sind z.B. Anforderungsspezifikation, Detailed Design, Maschinencode, Wartungsanleitung) sinnvollerweise gestaffelt. Dieses „Tailoring“ ist ausschlaggebend für eine Begutachtung, die eine vertretbare Balance zwischen Sicherheitsinteressen einerseits und terminlichen und kommerziellen Interessen andererseits bietet.

Die Dienstleistung „Begutachtung“ ist üblicherweise dem Markt unterworfen. Sie unterliegt – von Ausnahmen abgesehen – keinem Monopol.

4. Ausgewählte Aspekte

Einige Aspekte, die sich als entscheidend für die Qualität erwiesen haben, sind bei der Beschäftigung mit Software für sicherheitsrelevante Systeme in industriellen Anwendungen – oftmals in der Kerntechnik ebenso wie außerhalb der Kerntechnik – wiederholt aufgefallen. Eine Auswahl davon wird hier beschrieben.

4.1 Anforderungen an Qualität von Leittechnik und deren Software spezifizieren

Es passiert offenbar häufig, dass Software entwickelt wird, ohne die Vorstellungen von Qualität oder Sicherheit zuvor durchzusprechen oder gar zu spezifizieren. Am Ende der Entwicklung wird klar – sei es weil Kundenforderungen noch einmal durchgegangen werden, sei es weil eine Genehmigung aussteht – dass eine Qualitätsbestätigung fehlt. Der Maßstab, an dem die Qualität zu messen ist – z.B. die IEC 61508, eine andere Norm oder eine firmeninterne Vorgabe –, ist manchmal klar, aber häufig ist noch nicht einmal dieser Maßstab bekannt. Auf jeden Fall wurde die Software ohne (gute) Kenntnis des Qualitätsmaßstabes entwickelt.

Die IEC 61508 enthält mehr als 400 Einzelforderungen. Alleine diese Zahl macht deutlich, dass eine Konformität mit den Anforderungen vernünftigerweise nur erwartet werden kann, wenn die Anforderungen bei der Entwicklung explizit berücksichtigt und verfolgt wurden. Einige andere Standards enthalten noch deutlich mehr Forderungen.

Das für solche Entwicklungen die Begutachtung durchweg negativ ausfallen wird, ist offensichtlich. Das ist nicht nur für den Entwickler unerfreulich und wirtschaftlich schwierig. Auch für den Gutachter sind solche Situationen technisch und menschlich ärgerlich. Und sie lassen sich einfach vermeiden, indem bereits zu Projektbeginn die Anforderungen an die Qualität und Sicherheit und die dafür gültigen Maßstäbe (z.B. IEC 61508) durchgesprochen und ggf. berücksichtigt werden. Erfreulich und wirtschaftlich nützlich ist, dass dezidierte Kentechnik-Entwickler und eine zunehmende Anzahl konventioneller Entwickler diesen Weg beschreiten.

Die Qualitätsbestätigung muss nicht zwangsläufig jede Anforderung überprüfen. Sie kann sich auf eine Auswahl beschränken. Festgelegt wird das im Prüfplan.

4.2 Vorbenutzte Software

Sicherheitsrelevante Anwendungen werden häufig als Nischenanwendungen betrachtet, weil jede Technologie, z.B. Kernkraftwerke oder Eisenbahnen, für sich nur geringe Stückzahlen benötigt. Konsequenz daraus ist leider nicht die Entwicklung einer Anwendungstechnologie-unabhängigen Sicherheitsleittechnik mit dazugehöriger Software. Konsequenz ist vielmehr das Einbringen markgängiger Software in sicherheitsrelevante Bereiche.

Durchweg entspricht vorbenutzte Software den in Veröffentlichungen oder Standards gestellten Anforderungen an sicherheitsrelevante Software eher nicht. Die Gründe dafür wurden im Abschnitt 4.1 dieses Aufsatzes beschrieben. Die Abklärung, auf welche der üblichen Anforderungen bei vorbenutzter Software verzichtet werden kann, ist zeitaufwendig. Und ein für alle Beteiligten wirklich überzeugendes Ergebnis kommt kaum zustande. Deswegen ist die Neufassung der IEC 61508 zur vorbenutzten Software, wie Prof. Ehrenberger sie in diesem Tagungsband skizziert, zu begrüßen.

Trotz aller Schwierigkeiten ist es verschiedentlich Gelungen, vorbenutzte Leittechnik einschließlich deren Software für sicherheitsrelevante Anwendungen positiv zu begutachten.

4.3 Statische Analyse

Neben dem Software-Prüfverfahren „testen“ gibt es weitere Prüfverfahren, z. B. die Statische Analyse. Ihre Anwendung wird von der IEC 61508 mit gleicher Dringlichkeit verlangt wie das Testen. In der Praxis wird auf das testen sehr viel Aufwand verwendet

aber die statische Analyse bleibt meist komplett unberücksichtigt. Testen und Statische Analyse sind keine gleichgerichteten, untereinander austauschbaren Verfahrenen sondern sich ergänzende Verfahren mit unterschiedlichen Schwerpunkten. Mit dem Test wird gezeigt, dass der Code die geforderte Funktionalität erbringt. Die Statische Analyse weist auf potentielle handwerkliche Schwachstellen und unerwünschte (parasitäre) Funktionalität hin, z.B. das Teile des Hauptprogramms durch einen nicht sauber abgefangenen Interrupt aktiviert werden.

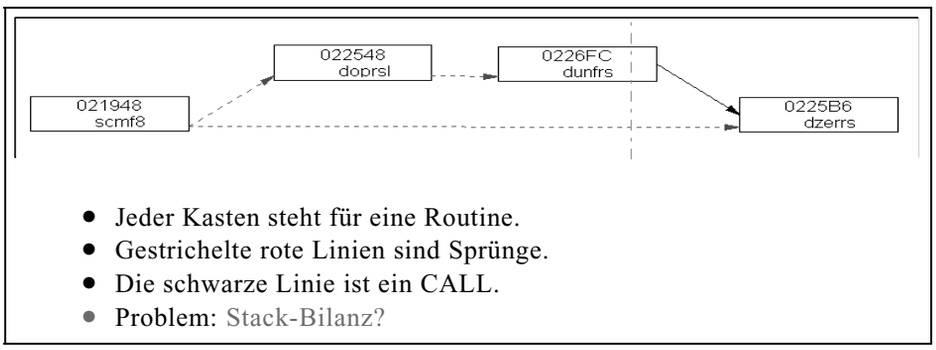

Bild 2: Ergebnis einer Statischen Analyse (Auszug; die gestrichelte vertikale grüne Linie ist hier ohne inhaltliche Bedeutung)

Wenn entsprechende Qualitätsmaßstäbe zugrunde gelegt werden, z. B. die IEC 61508, führt der Verzicht auf die Statische Analyse zu einer nicht regelkonformen Software. In der Konsequenz kann das dann zu Folgeproblemen bei der Begutachtung führen.

Auf jeden Fall führt der Verzicht auf die Statische Analyse zu einer wirtschaftlich ungünstigen Situation. Denn die Statische Analyse erfolgt meist werkzeuggestützt. Dadurch ist sie mit viel weniger Aufwand verbunden als der Test. Und sie hat das Potential systematisch Fehler zu entdecken, die zu den gefürchteten sporadischen Versagen von Software führen.

4.4 Zum Code

Seit der Zeit als Code noch die Form von Lochstreifen oder Lochkarten hatte, hat sich der Fokus in der Software-Entwicklung auf die frühen Phasen des Software-Engineering verlagert. Das ist auch für sicherheitsrelvante Software vernünftig. Unbenommen von dieser positiven Einschätzung und der vollen Würdigung der Bedeutung der frühen Entwicklungsschritte (siehe z.B. Abschnitt 4.1 dieses Aufsatzes) bleibt der für die Sicherheit von Anlagen ausschlaggebende Software-Teil der Maschinen-(Code).

Auf der Ebene des (Maschinen-)Codes beobachten wir seit wenigen Jahren Unzulänglichkeiten, wie sie noch in den achtziger Jahren undenkbar schienen. Dazu gehören:

- (unbenutzte) Interrupts, die nicht definiert belegt sind
- Flags, z.B. das Overflow-Flag, die nicht abgefragt werden
- fehlerbehaftete Routinen, die Compiler aus ihren Bibliotheken einbinden

Manche dieser „handwerklichen“ Unzulänglichkeiten können – das kann man systematisch nachvollziehen – zum sporadischen, nicht reproduzierbaren Versagen führen. Sie

werden offenbar auch mit sehr sorgfältigen und sehr aufwendigen Tests nicht entdeckt. In der Statischen Analyse werden sie sichtbar.

4.5 Die Schwarz-Weiß-Betrachtung bei Qualität und Sicherheit

Häufig resignieren Hersteller – gerade solche, deren Produkte respektable Qualität haben – angesichts der in Standards erhobenen Forderungen. Daher soll hier die Sicht auf Standards und Veröffentlichungen mit ihren Anforderungen an Software, deren Qualität, Zuverlässigkeit oder Sicherheit beleuchtet werden.

Man kann – und das geschieht leider sehr häufig – die Konformität mit Standards als binäre Situation auffassen:

- entweder die Software entspricht jeder einzelnen Forderung einer Norm voll und ganz
- oder sie entspricht der Norm eben nicht.

Diese Schwarz-Weiß-Betrachtung übersieht zwei wichtige Faktoren:

- Es wird kaum gelingen, ausgehend von einer weitgehenden Nicht-Erfüllung von für sicherheitsrelevante Software typischen Anforderungen in einem Projekt die weitgehende Erfüllung zu erreichen.
- Auch sehr hochwertige Software hat durchweg an der einen oder anderen (meist unbedeutenden) Stelle noch Eigenschaften, deren Konformität mit den gestellten Anforderungen nicht unmittelbar überzeugt.

Für den Weg hin zu besserer Software ist eine Betrachtung mit Zwischentönen nützlicher:

- Mit jeder Anforderung, die Software zusätzlich erfüllt oder die sie vollständiger erfüllt als bisher, wachsen Qualität und Eignung der Software für den Einsatz in sicherheitsrelevanten Anwendungen.

Selbstverständlich stellt sich bei einer solchen Betrachtung die Frage „Wieviel Konformität ist genügend Konformität?“. In einem Zertifizierungsverfahren wird die Antwort darauf im Bewertungsplan beschrieben.

Die IEC 61508 unterstützt die Betrachtung sicherheitsrelevanter Software mit Zwischentönen, indem sie im Teil 1, Abschnitt 4 (Conformance to this standard), den „degree of rigour“ einführt.

5. Werkzeuge

So wie die Entwicklung von Software heute nur noch mit Unterstützung leistungsfähiger Werkzeuge möglich ist, setzt auch die Qualitätsbestätigung (Begutachtung, Prüfung, Zertifizierung) leistungsfähige Hilfmittel voraus. Dazu gehören beim TÜV Nord:

- Ein praxisgerechtes Qualitätsmanagementhandbuch mit Vorlagen z.B. für Prüfpläne und Bewertungspläne.
- Die CATS Werkzeuge RiskCAT und RiskCAT_Railway, mit denen die Anforderungen der IEC 61508 und der EN 50128 für die Prüfplanung aufbereitet werden.
- Die CATS Statischen Analysatoren mit denen für verschiedene Prozessoren eine Statische Analyse des Maschinencodes erfolgen kann.

6. Schlußbemerkung

Die Qualitätsbestätigung (Begutachtung, Überprüfung, Zertifizierung) von sicherheitsrelevanter Software ist eine ingenieurmäßige Tätigkeit, die nach klar festgelegten Abläufen erfolgt. Sie kann deswegen in vorab übersehbaren Zeiten mit definierten Aufwänden durchgeführt werden.

Die häufig gestellte Frage „Und wer prüft den Gutachter?“ lässt sich für die Software&Elektronik Zertifizierungsstelle (SEECERT) und das Software&Elektronik Labor (SEELAB) des TÜV Nord einfach beantworten: Der Akkreditierer (DaTech) hat mit seiner Akkreditierung unsere Kompetenz bestätigt und kontrolliert uns regelmäßig.

Literaturverzeichnis

[VDI93] VDI-Gemeinschaftsausschuß Industrielle Systemtechnik; Software-Zuverlässigkeit; VDI Verlag, 1993