SOFTWARE-ENTWICKLUNG + UNTERSTÜTZENDE TOOLS = SICHERE MEDIZINGERÄTE?

Gudrun Neumann, SGS-TÜV Saar GmbH

Jürgen Triep, QA Systems GmbH

SGS TÜV SAAR

WHEN YOU NEED TO BE SURE

SGS

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel

EINFÜHRUNG

Was ist ein Software Tool?

- Ein Software Tool ist ein Programm zur Unterstützung der Softwareentwicklung.

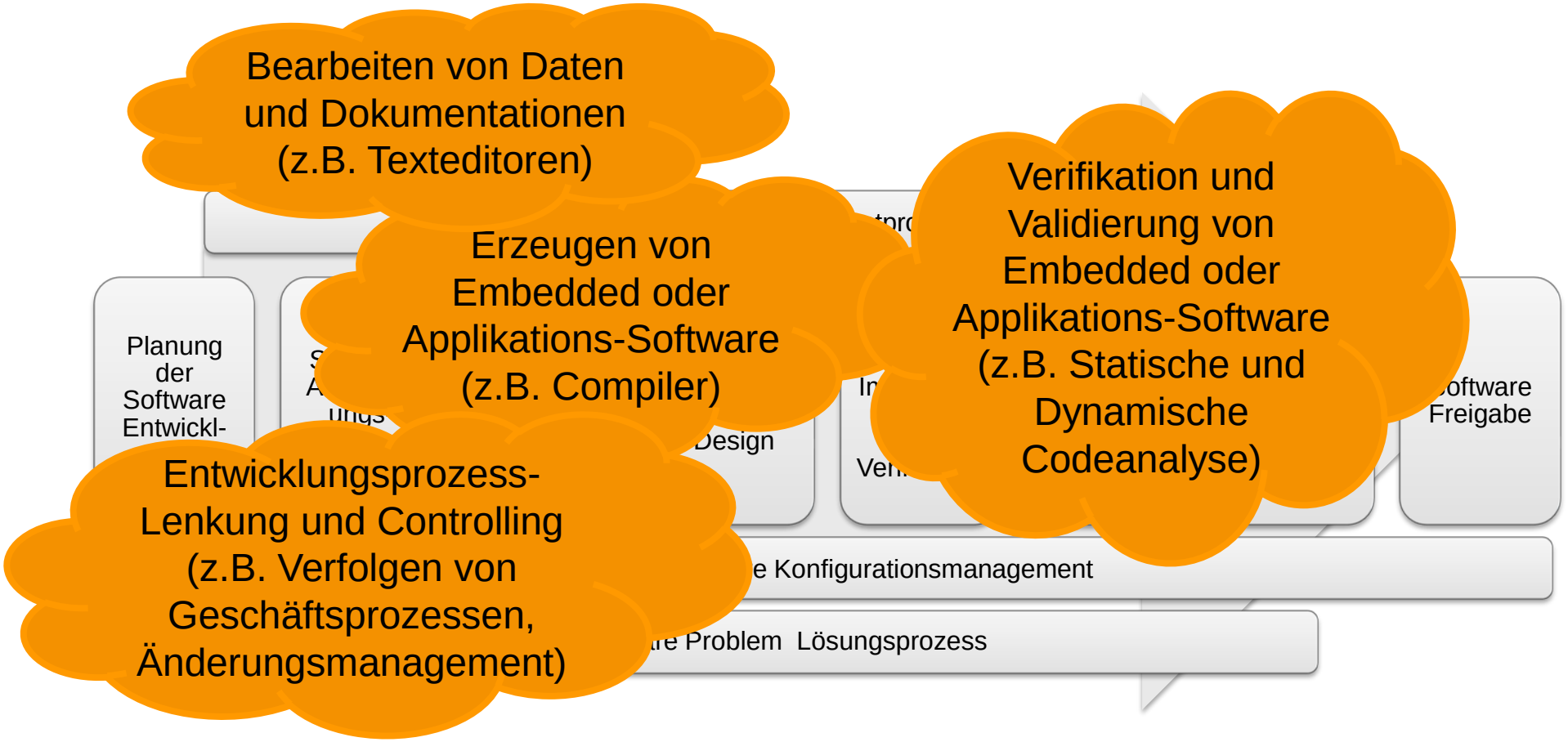
Software -- Funktionale Sicherheit – Beispiele:

- 1983: Mehrere Tote bei medizinischen Bestrahlungen in den USA
  - Bediener hatten Tastatur zu schnell bedient
- 1993: Airbus-Unfall in Warschau
  - Software kannte Aquaplaning nicht

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel

WELCHE TÄTIGKEITEN UND AUFGABEN WERDEN VON SW-TOOLS UNTERSTÜTZT?



Der gesamte Entwicklungsprozess kann toolunterstützt durchgeführt werden.

FEHLER DIE DURCH TOOLS ZUR SW ENTWICKLUNG ENTSTEHEN KÖNNEN

- Daten, z.B. Anforderungen, gehen verloren oder werden verfälscht
- Fehlerhafte Erzeugung von ausführbarer Software oder Codegenerierung
- Fehler in ausführbarer Software werden nicht gefunden
- Fehlerhafte Parametrisierung, Konfiguration
- Bei Aufbau einer Toolkette werden nicht geeignete Tools kombiniert, dadurch entstehen komplexe Schnittstellen

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel

ALLGEMEINE ANFORDERUNGEN AN TOOLS

- Toolverhalten bzw. -ergebnis entspricht der Spezifikation
- Ergebnisse sind zuverlässig reproduzierbar
- Einfache verständliche Bedienoberfläche
- Einweisung in die Bedienung, falls notwendig
- Verständliches Handbuch
- Serviceverfügbarkeit
- Klassifikation bzw. Qualifikation des Tools

VORGEHENSWEISE BEI EINER TOOL KLASSIFIKATIONEN

1. Feststellen der Anwendungsfälle
2. Pro Anwendungsfall wird der mögliche Einfluss des Tools auf das Endprodukt analysiert, man unterscheidet:
   - Tools die **keinerlei Einfluss** auf das Produkt haben, wie z.B. ein Tool zur Präsentation von Geschäftszahlen
   - Tools die **indirekten Einfluss** auf das Produkt haben, wie z.B. ein Verifikationstool
   - Tools die **direkten Einfluss** auf das Produkt haben, wie z.B. ein Compiler
3. Definieren von Maßnahmen, um „Tool“-Fehler aufzudecken und zu beherrschen
4. Einordnung des Tools in eine Klasse entsprechend des geforderten Standards
5. Dokumentation der Ergebnisse

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel

BEISPIEL VERIFIKATIONSPHASE

Herausforderungen in der Entwicklung:

- Auswahl geeigneter Testfälle ist oft schwierig.
- Bei komplexen Systemen sind eine Vielzahl von Testfällen notwendig, um eine ausreichende strukturelle Abdeckung zu erreichen.
- Dokumentation der Testergebnisse entsprechend den Anforderungen aus den Standards.

STRUKTURABHÄNGIGE SOFTWARE TESTS (STRUCTURAL TEST COVERAGE)

- Entsprechend IEC 61508-3:2010, Tabelle B.2 “Dynamic analysis and testing”:

	Technique/Measure *	Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Test case execution from boundary value analysis	C.5.4	R	HR	HR	HR
2	Test case execution from error guessing	C.5.5	R	R	R	R
3	Test case execution from error seeding	C.5.6	---	R	R	R
4	Test case execution from model-based test case generation	C.5.27	R	R	HR	HR
5	Performance modelling	C.5.20	R	R	R	HR
6	Equivalence classes and input partition testing	C.5.7	R	R	R	HR
7a	Structural test coverage (entry points) 100 % **	C.5.8	HR	HR	HR	HR
7b	Structural test coverage (statements) 100 %**	C.5.8	R	HR	HR	HR
7c	Structural test coverage (branches) 100 %**	C.5.8	R	R	HR	HR
7d	Structural test coverage (conditions, MC/DC) 100 %**	C.5.8	R	R	R	HR

NOTE 1 The analysis for the test cases is at the subsystem level and is based on the specification and/or the specification and the code.

NOTE 2 See Table C.12.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

** Where 100 % coverage cannot be achieved (e.g. statement coverage of defensive code), an appropriate explanation should be given.

EINFACHES BEISPIEL

```
int fct_example (int x, int y)
 {
       int z = 0;
       if ((x>0) && (y>0)){
             z = x; }
       return z;
 }
```

‘fct_example’ ist Teil einer größeren Anwendung und diese wird mit einer Testsuite getestet.

- Funktion ‘fct_example' wurde mindestens einmal während des Tests aufgerufen: „Function Coverage“ ist 100% bzgl. dieser Funktion.
- Die Funktion wurde aufgerufen als ‘fct_example (1,1)’: „Statement Coverage“ ist 100% bzgl. dieser Funktion.
- Die Funktion wurde aufgerufen als ‘fct_example (1,1)’, ‘fct_example (0,1)’, ‘fct_example (1,0)’ und ‘fct_example (0,0)’: „Condition/Decision Coverage“ ist 100% bzgl. dieser Funktion.

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel

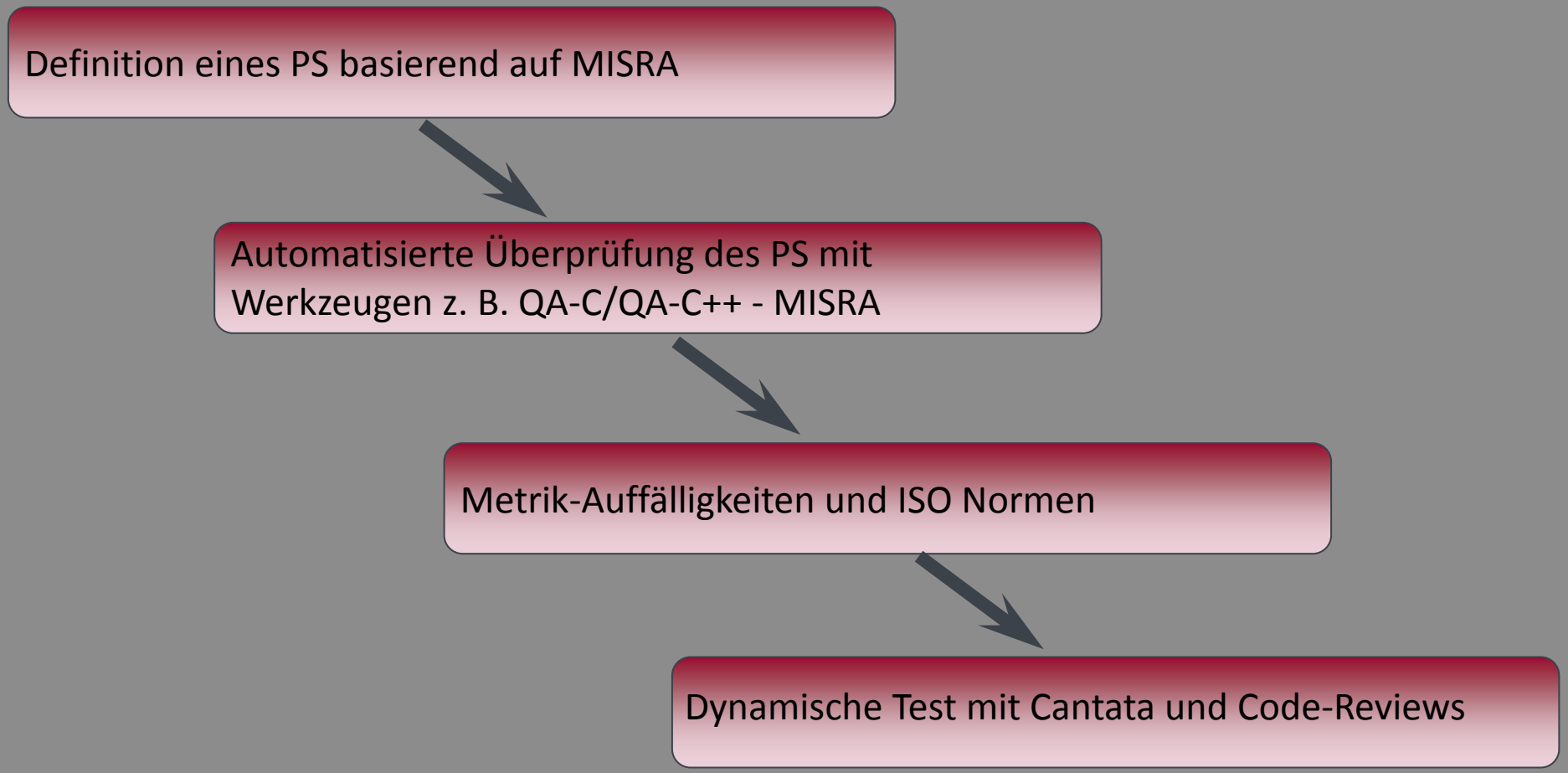
FAZIT

Tools erleichtern die Entwicklung von Software, wenn entsprechende Anwendungsregeln beachtet werden:

- Auswahl der Tools entsprechend der Problemstellung.
- Anwender sollten für die verwendeten Tools geschult werden.
- Liste bekannter Toolfehler beachten und vermeiden.
- Für sicherheitsrelevante Software sind zusätzliche Maßnahmen zu beachten, je nach Anwendungsfall und Sicherheitsanforderung.
- Von den Standards (z.B. IEC 62304) wird toolunterstützte Entwicklung gefordert.
- Klassifikation bzw. Qualifikation der Tools wie von den Standards gefordert.

AGENDA

- Einführung
- Einfluss von Tools in der Softwareentwicklung
- Anforderungen an Tools
- Beispiel Verifikationsphase
- Fazit
- Praxisbeispiel



DANKE FÜR IHRE AUFMERKSAMKEIT

SGS-TÜV Saar GmbH

Functional Safety

Hofmannstrasse 50

D-81379 Muenchen

Germany

www.sgs-tuev-saar.com/fs

QA Systems GmbH

The Software Quality Company

Schwieberdinger Str. 56

70435 Stuttgart

Germany

www.qasystems.de

Gudrun Neumann

Product Manager Functional Safety Software

Industrial Functional Safety Expert IFSE

Automotive Functional Safety Professional AFSP

E-Mail: gudrun.neumann@sgs.com

Telefon: +49 89 787 475 -216

Fax: +49 89 787 475 -217

Jürgen Triep

Geschäftsführer

E-Mail: jtriep@qasystems.de

Telefon: +49 (0)89 898606 14

Fax: +49 (0)89 898606 15