



Angepasste Software Standards für DLR- Eigenentwicklungen

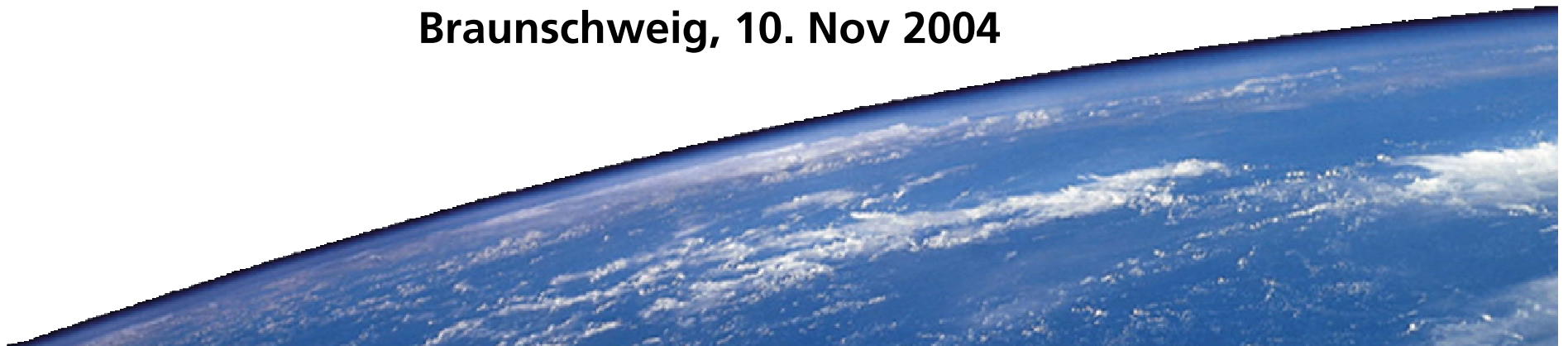
- Die DLR Software Basisstandards -

Anita Herrmann

<Anita.Herrmann@dlr.de>

Simulations- und Softwaretechnik

Braunschweig, 10. Nov 2004





Ausgangspunkte

Im DLR werden nach vorsichtigen Schätzungen jährlich ca. 50 % der Personalkosten in Softwareentwicklungen investiert.

Bisher gibt es keine definierten Grundregeln, die bei der Entwicklung von Software zu beachten sind.

- ⇒ Die Qualität der Software entspricht dem Informatikwissen der Entwickler (zumeist keine Fachleute).
- ⇒ Qualitäts-Defizite führen zu **späterem Mehraufwand** bei Wartung, Weiterentwicklung und evtl. Wiederverwendung.

2002: Beschluss des Vorstandes zur Entwicklung von Software Basis-Standards, integriert in das DLR Intranet.

Die DLR Software Basis-Standards haben das Ziel, den Investitionsschutz der Software Entwicklungen im DLR zu unterstützen.



Software Investitionsschutz - Ziele

- **Effektive Verwendung der Ressourcen während der Entwicklung und Weiterentwicklung der Software**
- **Bewahrung des Wertes der entwickelten Software.**
- **Schutz gegen Haftungsforderungen, die aus evtl. Fehlern der Software im Betrieb resultieren.**

Noch nicht einbezogen:

- **Software-Patentschutz; Schutz von Software-Bezeichnungen, Marken und Algorithmen, Verwertungs- und Nutzungsrechte;**

Nicht anzuwenden für:

- **Ein-Zweck Software, nicht weiter verwendbare SW-Prototypen;**
- **wenn gesetzliche bzw. Auftraggeber - Forderungen umfassender und inhaltlich anders strukturiert sind (z.B. Anwendung V-Modell des Bundes).**



Investitionsschutz-Grundanforderungen

1. Eine geprüfte und von der übergeordneten Leitung/Auftraggeber genehmigte Aufgabenstellung zur Software-Entwicklung.
2. Kontinuierliche Überwachung des Entwicklungsfortschrittes/-risikos, Einleitung von Vorbeugungsmaßnahmen bei absehbaren Problemen.
3. Die Festlegung der Hauptfunktionalitäten der Software und eine dafür adäquate Aufwands- und Zeitkalkulation.
4. Anwendung Fehler vermeidender und von Code-Lesbarkeits-Regeln in den Software Produktionsphasen (Design und Codierung).
5. Durchführung und Nachweis von systematischen Tests zur Einsatzfähigkeit der Software.
6. Maßnahmen zur Sicherung der Entwicklungsergebnisse und Nachweis der Problembeseitigung (Minimum: Versions-Directory, tabellarische Problemauflistung, Back-up).
7. Zumindest eine dokumentierte Anweisung für die Installation und Nutzung der Software.



Prinzipien der DLR Software Basis-Standards

- ▶ **Minimale Anforderungen durch Tailoring**
 - Je nach Art und Kontext der Software werden gezielt unterschiedliche Einzelmaßnahmen des Investitionsschutzes durch das System vorgeschlagen.

- ▶ **Stufung der Notwendigkeit dieser Maßnahmen**
 - „M“ – mandatory (unverzichtbare Grundanforderungen),
 - „R“ – recommended,
 - „D“ – desirable (nur bei ECSS-Anforderungen).

- ▶ **Ergänzende Informationen und Hilfen zur Umsetzung der vorgeschlagenen Maßnahmen (Wissens-Links).**

- ▶ **Unterstützung der Vorbereitung von Angeboten durch verlässliche Interpretation der Standards beim Tailoring (ECSS).**

- ▶ **Teilautomatische Generierung von Projekt-Dokumenten (z.B. SW QS-Plan, Projekt-Entwicklungsplan), down-load Nutzung vorhandener Vorlagen.**



DLR Software Basis-Standards – Status 11/2004

⇒ **Software/projektspezifisch angepasste Investitionsschutz-Anforderungen können derzeit entsprechend**

- den DLR internen Software Minimal-Standards oder nach
- ECSS-E-40 und ECSS-Q-80 (Raumfahrt)
- RTCA/DO-178B (Luftfahrt)
- EN 50128 (Software für Bahntechnik).

online über das DLR-Intranet ausgegeben werden.

⇒ **Anhand von realen DLR Software Projekten umfassend getestet**

⇒ **bis Ende 2004/2005:**

Einbindung von Wissens-Links zu den Investitionsschutz-Anforderungen im Feld „Umsetzungsempfehlungen“; Dokumentengenerierung; Komplettierung des Handlings, Zugriffsschutz.

⇒ **Nachfolgende Demonstration derzeit verfügbarer Funktionen noch in der Testumgebung**



Sicherung und Bewahrung von Entwicklungsergebnissen – Beispiel 1

5 Sicherung und Bewahrung von Entwicklungsergebnissen				
5.1.0.0	Mindestanforderungen an den Investitionsschutz von Software bei der Sicherung und Bewahrung von Entwicklungsergebnissen	M	QM-Plan, KMP, SVVP	SRR, PDR, CDR, QR, AR, ORR
5.2.2.0	Mindestanforderungen an das Konfigurationsmanagement von Software-Prototypen	M		
5.2.6.0	Anwendung von systematischem Back-up	M	Konfigurationsobjekte, Konfigurationsdatei	laufend
5.3.0.0	Notwendige aufzuzeichnende Informationen über Probleme und Änderungsanforderungen	M	Nonconformance Report/Change Request, MF	laufend
5.3.2.0	Überwachung der Bearbeitung von Korrekturen und Änderungen bei kleinen Projekten	R	Nonconformance Report/Change Request, MF	laufend
5.3.3.0	Rückverfolgbarkeit von Änderungen im Sourcecode	M	Sourcecode	KM-Eingangsprüfung
5.6.4.0	Kennzeichnung von Software auf externen Speichermedien	M	Konfigurationsobjekte, Konfigurationsdatei	AR, ORR, Release-Freigabe
6 Software Verifikation und Validation				
6.1.1	Mindestanforderungen an die Verifikation und	M		AR

Beispiel: Neu-Entwicklung eines **Software – Prototypen (Software zur Klärung wissenschaftlicher Aufgabenstellungen mit evtl. späterer Weiterverwendung)**

⇒ **Abstriche bei den Anforderungen an die Spezifikation, Design, Verifikation, Test, Dokumentation, Konfigurationsmanagement.**

⇒ **Keine Abstriche bei den Anforderungen an das Projektmanagement, Fehlervermeidung bei der Implementierung, Code-Kommentierung**



Sicherung und Bewahrung von Entwicklungsergebnissen – Beispiel 2

4.4.6.1	Minimum des Tests von unkritischen Komponenten	M	Sourcecode, DJF	KM-Eingangsprüfung
5 Sicherung und Bewahrung von Entwicklungsergebnissen				
5.1.0.0	Mindestanforderungen an den Investitionsschutz von Software bei der Sicherung und Bewahrung von Entwicklungsergebnissen	M	QM-Plan, KMP, SVVP	SRR, PDR, CDR, QR, AR, ORR
5.2.0.0	Softwareobjekte, die unter Konfigurationskontrolle zu stellen sind.	M	KMP, Konfigurationsdatei	SRR, PDR, CDR, QR, AR, ORR
5.2.3.0	Anwendung der Grundsätze des Versionsmanagements von Software und Systemen	R	KMP, Konfigurationsdatei	PDR, CDR, QR, AR, ORR
5.2.4.0	Identifikation von Konfigurationsobjekten	R	KMP, Konfigurationsdatei	SRR, PDR, CDR
5.2.5.0	Gewährleistung der Konsistenz und Qualität der Konfiguration	M	KMP, Konfigurationsdatei	SRR, PDR, CDR, DDR, QR, AR
5.2.6.0	Anwendung von systematischem Back-up	M	Konfigurationsobjekte, Konfigurationsdatei	laufend
5.3.0.0	Notwendige aufzuzeichnende Informationen über Probleme und Änderungsanforderungen	M	Nonconformance Report/Change Request, MF	laufend
5.3.1.0	Vefahren zur systematischen Erfassung, Bearbeitung, Verfolgung und Information der aufgetretenen Probleme und Änderungsanforderungen im Projekt.	M	Nonconformance Report/Change Request, MF	laufend
5.3.1.1	Überprüfung des Designs bei der Wartung und Weiterentwicklung der Software.	R	Nonconformance Report/Change Request, MF	laufend

Beispiel: Wartung/Weiterentwicklung eines **industriell angewendeten SW-Codes** (z.B. FLOWer, TAU)

- ⇒ **Keine** Anforderungen an die Neuspezifikation der Software, Architektur -Design, CASE, Auswahl von Compilern;
- ⇒ **Erhöhte** Anforderungen an das **Konfigurationsmanagement**, Designüberwachung, Korrektheit der Auslieferung an Kunden;
- ⇒ **Keine Abstriche** beim Projektmanagement, Design- und Nutzerdokumentation, Fehlervermeidung bei der Codierung;
- ⇒ **Modifiziertes** Verifikations-, Testmanagement (gegenüber Neuentwicklung).



Nutzung im DLR Intranet

- ▶ **Dezentrale Nutzung des Systems durch die Projektverantwortlichen bzw. Software-Entwickler als Werkzeug und Informationsmittel.**
- ▶ **Einführung 2005 im Rahmen des DLR Bildungsprogrammes:
Seminar „Einführung in das Software Engineering / die Software Qualitätssicherung im DLR“**
- ▶ **Verfügbar ab 03/2005 im neuen DLR Intranet**