

Software-Engineering in KMU – vom kreativen Chaos zum Qualitätsmodell

Dr. Roland Feindor, FH Rosenheim



Prof. Dr. Roland Feindor

Software-Entwicklung (SE) in kleinen und mittleren Unternehmen (KMU): - vom kreativen Chaos zum Qualitätsmodell



Inhalt

- **Kleine und mittlere Unternehmen (KMU)**
 - Was ist ein KMU?
 - Bedeutung der KMU allgemein und in der Software-Entwicklung
- **Software-Engineering in KMU**
 - Projekt SE-KMU
 - Praktische Erfahrungen
- **Vom kreativen Chaos zum Qualitätsmodell**
 - Leitfaden und SE-Check
 - Klassische Methoden (CMMI, V-Modell, agile Methoden)
 - Open-Source Tool-Kette



Kleine und mittlere Unternehmen (KMU)

Typ	Beschäftigte	Umsatzerlös oder Bilanzsumme (Mio €)	
Mittlere Unternehmen	< 250	≤ 50	≤ 43
Kleine Unternehmen	< 50	≤ 10	≤ 10
Kleinst-unternehmen	< 10	≤ 2	≤ 2



Bedeutung der KMU

Unternehmens- kategorie	Mitarbeiterzahl	Anteil an allen Unternehmen
mittel	50 - 249	3,0%
klein	10 - 49	13,7%
kleinst	1 - 9	82,7 %
Summe KMU	1 - 249	99,5 %



Software – Entwicklung in KMU

- Ein sehr großer Teil der Software-Entwicklung findet in kleinen und kleinsten Software-Häusern statt.
- Nach Untersuchungen in Deutschland haben über 75 % der Software-Unternehmen weniger als 10 und über 90% weniger als 50 Mitarbeiter.
- Viele Software-KMU sind auf spezielle Nischen spezialisiert.



Projekt SE-KMU

- Gefördert durch die Bayerische Forschungsstiftung
Laufzeit: 1.3.2006 – 31.10.2008
- Aktive Mitwirkung von 7 Software-KMU



CREATIV.CONSULTING.GMBH





Ziele des Projektes SE-KMU

- Wie entwickeln KMU Software?
- Was unterscheidet deren Vorgehensweise von Großunternehmen?
- Welche Stärken und Schwächen zeigen sich?
- Wie kann man die Software-Engineering-Prozesse in KMU konkret verbessern?



Einige Schlaglichter aus Software-KMU

- Prozesse
 - Prozesse gelebt, gewachsen, (erfolgreich)
 - Prozesse kaum dokumentiert, Wissen in den Köpfen
 - kaum Dokumente während der Projektdurchführung
 - Projektergebnis: Laufende Software + Benutzerdoku
 - Schwerpunkt: Persönliche Kommunikation
- Werkzeug-Einsatz sehr unterschiedlich
 - zum Teil nur IDE ohne Versionsmanagement
 - typische Werkzeuge wie Issue-Tracker teilweise unbekannt
 - kaum automatisierte Tests, automatisierte QS, autom. Build
 - kaum Modellierung, kaum Spezifikation, „nur“ Codierung
 - zum Teil umfassende Werkzeugunterstützung (z.B. MS-Team Foundation Server)



Einige Schlaglichter aus Software-KMU

- Ursachen für unterschiedliche Vorgehensweisen
 - geringere Projektgrößen
 - spezielles Domänenwissen
 - geringerer Kommunikationsbedarf durch weniger Schnittstellen
 - geringerer Dokumentationsbedarf bei kleineren Auftraggebern

- Stärken (Beispiele)
 - flache Hierarchien und enger Teamzusammenhalt
 - wenig Bürokratie
 - enger Kundenkontakt durch Entwickler
 - direkte Verantwortung der Geschäftsleitung für den Erfolg
 - direktes Feedback bei Fehlern und Mängeln der Software
 -



Einige Schlaglichter aus Software-KMU

- Schwächen (Beispiele)
 - eingeschränkte Ressourcen
 - Abhängigkeit von einzelnen Entwicklern
 - verbesserungswürdiges Qualitätsmanagement
 - zu wenig Dokumentation
 - kaum Forschungsmöglichkeiten
 -



Warum sollten KMU ihre Vorgehensweise in der Software-Entwicklung verbessern?

- **Effizienteres** Arbeiten
 - Verbesserung der Kommunikation
 - Unterstützung von Routinetätigkeiten durch Werkzeuge
 - Vorlagen für Projekte / Dokumente / Angebote / Software
 - Effizienteres Vorgehen / Praktiken / Methoden
(z.B. weniger Fehler durch kontinuierliche Integration)
- **Zuverlässigere Planung**, z.B. zur Aufwandsschätzung
- **Nachvollziehbarkeit**
 - Geringere Abhängigkeit von Mitarbeitern
 - verbesserte Qualität
- **Transparenz**



Leitfaden “Software-Entwicklung in KMU”

Basis: SWEBOK (Guide to the Software Engineering Body of Knowledge, 2004)

www.swebok.org

2. Software Requirements

3. Software Design

4. Software Construction

5. Software Testing

6. Software Maintenance

Phasen (= Chapter 2-6) und
unterstützende Prozesse (7 -12)

7. Software Configuration Management

8. Software Engineering Management

9. Software Engineering Process

10. Software Engineering Tools and Methods

11. Software Quality

12. Knowledge Areas of the Related Disciplines



Software-Engineering-Check

- Prüfen des Vorgehens entsprechend SE-Leitfaden
- Varianten nach Projektart
 - S: Standard-Software
 - S/I: Modul-Software
 - I: Individual-Software
- Empfehlungen zum Einsatz (angelehnt an EN 50128:2001)
 - V: verbindlich
 - DE: dringend empfohlen
 - E: empfohlen
- Einbindung von Management-Anforderungen
 - CMMI (level 2)



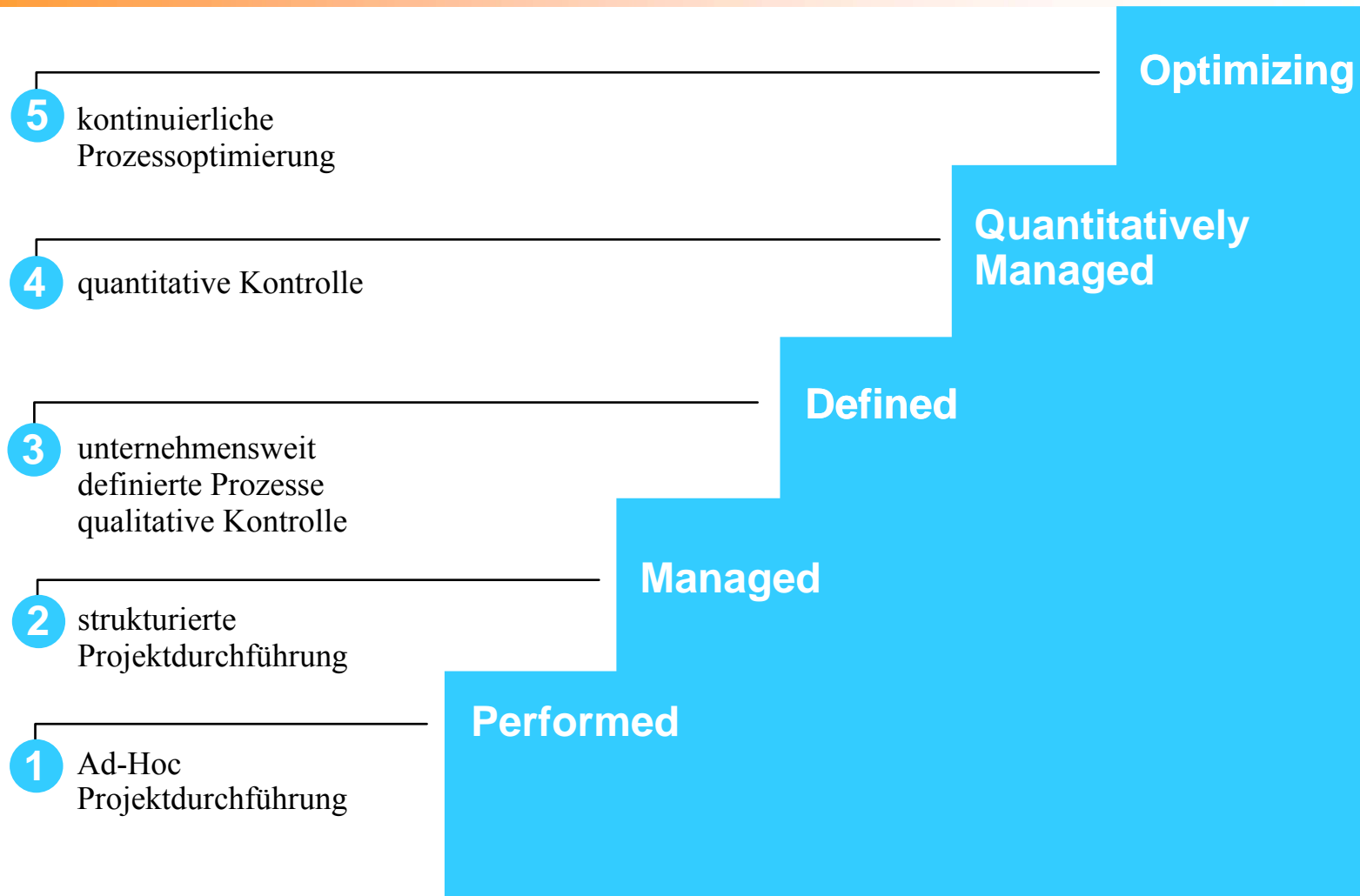
Software-Engineering-Check

1	2	3	4	A	B	C	D	E	F	G	M	N	O	P	Q	R	S	
				Kap./Id	Abschnitt	Beispiele / Kriterien / Ergebnisse	S	S/I	I	CMMI	Referenz SWEBOK	Referenz CMMI	Kriteriums- gewicht	IST	Nutzwert IST	SOLL	IST- SOLL	
				SE-CHECK V3.0													Projektart <input type="checkbox"/> I CMMI <input type="checkbox"/> Nein	
				Anforderungsphase													10,22	
				Kundenanforderungen entwickeln														
				1.1a	Beachten aller Prozessbeteiligten		V	V	V	-	-	RD SP 1.1, 1b	1,03	0,00	0,00	1,00	-1,03	
				21	1.1b Einbeziehen unterschiedlicher Anforderungsquellen		V	V	V	V	2.3a	RD SP 1.1, 1b	1,03	0,00	0,00	1,00	-1,03	
				28	1.1c Verwenden unterschiedlicher Bestimmungstechniken		DE	DE	V	V	2.3b	RD SP 1.1, 1b	1,03	0,00	0,00	1,00	-1,03	
				34	1.1d Besichtigung der Systemumgebung		E	V	V	V	-	RD SP 1.1, 1b	1,03	0,00	0,00	1,00	-1,03	
				Produktanforderungen entwickeln														
				40	1.2a Anforderungen an Produkt und Komponenten aufstellen und klassifizieren		V	DE	DE	V	-	RD SP 2.1, 2b	0,52	0,00	0,00	1,00	-0,52	
				46	1.2b Anforderungen an Schnittstellen definieren		V	V	V	V	-	RD SP 2c	1,03	0,00	0,00	1,00	-1,03	
				Anforderungen analysieren														
				51	1.3a Konzeptionelle Modellierung		V	V	V	V	2.4b	RD SP 3a	1,03	0,00	0,00	1,00	-1,03	
				57	1.3b Machbarkeitsanalyse durchführen		DE	E	E	-	-	-	0,21	0,00	0,00	1,00	-0,21	
				59	1.3c Systemdefinitionsdokument erstellen		E	E	E	V	2.5a	RD SP 3b	0,21	0,00	0,00	1,00	-0,21	
				61	1.3d Analyse und Abgrenzung der Anforderungen		V	V	V	V	-	RD SP 3.3, 3d	1,03	0,00	0,00	1,00	-1,03	
				Anforderungen validieren														
				66	1.4a Anforderungsreviews durchführen		DE	DE	DE	V	2.6a	RD SP 3e	0,52	0,00	0,00	1,00	-0,52	
				69	1.4b Prototypen erstellen		E	E	DE	V	2.6b	RD SP 3e	0,52	0,00	0,00	1,00	-0,52	

SE-KMU: vom kreativen Chaos zum Qualitätsmodell



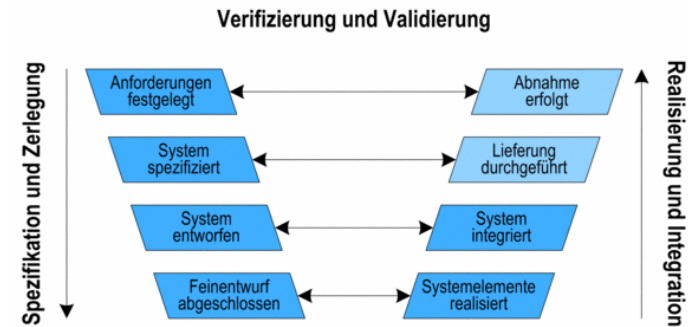
Klassische Methoden: CMMI





Klassische Methoden: V-Modell XT

- Vorgehensmodell zum Planen und Durchführen von Projekten
- Regelt: Wer? Wann? Was?
- Ziele:
 - Projektrisiken minimieren
 - Qualität verbessern und gewährleisten
 - Gesamtkosten eindämmen (Projekt- und Systemlebenszyklus)
 - Kommunikation zwischen allen Beteiligten verbessern
- Merkmale:
 - Produktzentriert
 - Kooperation Auftraggeber und Auftragnehmer geregelt
 - Tailoring: Anpassung an Projekt
 - Schwerpunkt Management-Mechanismen, nicht Methodik





„Klassische“ Methoden: agile Entwicklung

Manifesto for Agile Software Development (2001)

Hoher Wert:

- *Prozesse und Werkzeuge*
- *umfassende Dokumentation*
- *Vertragliche Verhandlung*
- *Einhaltung eines Planes*

Wertvoller:

- *Menschen und Zusammenarbeit*
- *Funktionierende Software*
- *Zusammenarbeit mit dem Kunden*
- *Reaktion auf Veränderung*



Agile Entwicklung: unsere Empfehlung

Reinhold-Messner-Stil

- *ohne Expeditionsballast*
- *mit bewährter Ausrüstung*
- *in kleinem Team*
- *mit klarem Plan*
- *mit rascher Durchführung*
- *mit ständiger Kontrolle*
- *mit notwendigen Änderungen*





Tool-Unterstützung in der Software-Entwicklung

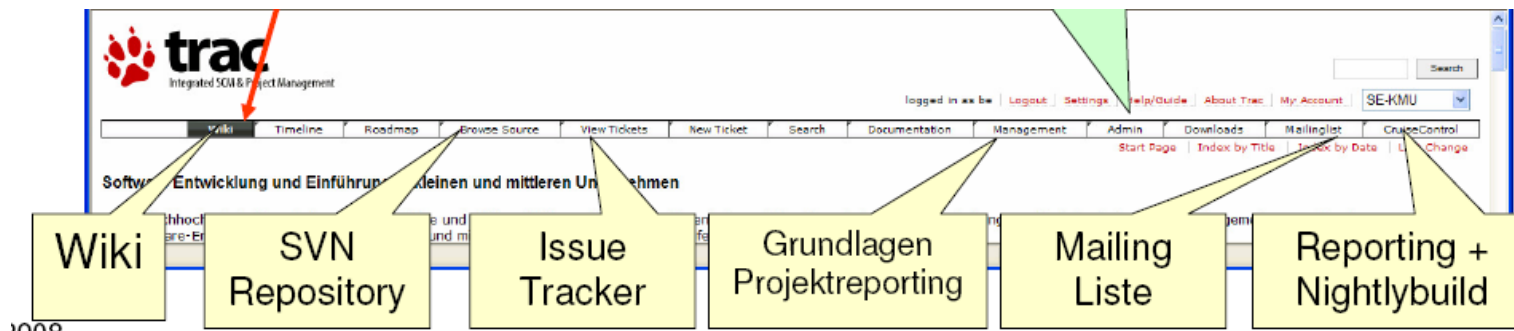
Was braucht ein (kleines) Projekt mindestens?

- Mailverteiler
- Gruppenlaufwerk / Projektablage
- Projekthandbuch / Projekt-Wiki
- Versionsmanagement / Konfigurationsmanagement
- Issue-Tracker
- Automatisiertes Build / Continuous Integration Infrastruktur
- Möglichkeit für Projektberichte und -analysen

Ziel: Neue Projekte „per Knopfdruck“



Tool-Unterstützung – Beispiele im Java-Umfeld



- Prüfspezifikation Systemelement
=> JUnit Test Case
- Prüfprotokoll Systemelement
=> JUnit Report
- Änderungsanforderung / -bewertung
=> Trac / Issue Tracker
- Konfigurationsmanagement => SVN, CVS



Vielen Dank für Ihre Aufmerksamkeit



Haben Sie noch Fragen?