

# Six Sigma in der Software- Entwicklung (?)

Martin Mikusz

Universität Stuttgart

Lehrstuhl für Allgemeine Betriebswirtschaftslehre und  
Wirtschaftsinformatik II (Unternehmenssoftware)

Prof. Dr. Georg Herzwurm

{herzwurm;mikusz}@wi.uni-stuttgart.de



# Six Sigma

- In den 80 Jahren bei Motorola als ein auf statistischen Methoden basierender Prozessverbesserungsansatz entwickelt
- Six Sigma Projekte folgen einer formalisierten Vorgehensweise
  - Projektphasen: Define, Measure, Analyze, Improve und Control (DMAIC)
  - Phasenspezifische Methoden, Zwischenergebnisse und Meilensteine
  - Definierte Rollen (mit Fortbildung v. a. in Statistik und im Projektmanagement)
    - Black Belts: Projektinitiator, -leiter und Change Manager (Vollzeit)
    - Green Belts: Unterstützen Black Belts bei der operativen Ausführung von Six Sigma Projekten (50%)
- Vorgehensweise in aller Kürze
  - Problem einem Prozess zuordnen - Prozess beschreiben - Haupteinflussgrößen auf den Prozess finden und messen - Problemursache mit statistischen Methoden finden - Prozess verbessern - Verbesserung institutionalisieren
  - ... „Eingebettet“ in striktes Projektmanagement



# Problemstellung und Vorgehensweise

- Philosophie und Konzeption von Six Sigma noch nicht in jeder Hinsicht geklärt [1]
  - Unzählige Differenzierungsbemühungen in der Praxis
    - Ansätze: Lean 6 $\sigma$ , Ultimate 6 $\sigma$ , 6 $\sigma$  Gen III, 6 $\sigma$  Plus,  $\sigma$  Lean, etc.
    - Vorgehensweisen: DMAIC, DMADI, DMADV, DMADVI, DMAIC etc.
    - Methoden: Über 450 potentielle Six Sigma-Methoden [2]
    - Einsatzspektrum: Industrielle Produktion  $\rightarrow$  Dienstleistungssektor  $\rightarrow$  Softwareentwicklung
    - Einsatzphase: Produktion  $\rightarrow$  Entwicklungsprozesse (Design for Six Sigma)
    - Zielsetzung: Qualitätsinitiative mit dem Ziel einer nahezu fehlerfreien Produkt- und Prozessqualität  $\rightarrow$  Umfassender Business Excellence-Ansatz
  - Notwendige empirische Behandlung von Six Sigma nicht gegeben [3]
- $\rightarrow$  Fokus auf *Prinzipien*, welche den Six Sigma Ansätzen unterliegen, für die Fragestellung Ziel führend [ff. ähnlich bei [4]]:
- Variation der qualitätskritischen Steuerparameter
  - Metriken und datenbasierte Entscheidungsfindung
  - Statistische Analysen als Erkenntnisweg
  - Statistische Prozessregelung
- $\rightarrow$  Fokus auf Anpassung von Six Sigma über domänen- und problemspezifische Analogien oder bereits vorhandene Ansätze in der Softwareprozess-Verbesserung



## Variation der qualitätskritischen Steuerparameter

- Six Sigma konzentriert sich auf die Reduzierung von Variation bei den qualitätskritischen Steuerparametern, die das Ergebnis stark beeinflussen
- In der Software-Entwicklung sind dies (bez. Produktivität der Softw.-Entw; im ff. [5]):
  - Prozesscharakteristika: v. a. Unternehmenskultur 1:11; Mitarbeiterqualifikation 1:10; Wiederverwendung 1:4
  - Produkthanforderungen: v. a. Produktkomplexität 1:2,36
- Varianzreduktion möglich, aber nicht mit statistischen Methoden von Six Sigma
- Gleichzeitig überdecken diese Einflussfaktoren die Problemursachen und Effekte der Prozessverbesserung bei den für Six Sigma zugänglichen Problemstellungen
  - Bsp.: Moderate Erhöhung der Testeffizienz durch CASE-Tools schwer nachweisbar, da überlagernde Produktivitätsschwankungen durch Mitarbeiterqualifikation u. U. „1:10“!
- Variation in der Summe enorm: Bei einem „typischen“ Entwicklungsprojekt von ca. 80.000 LOC variiert [im ff. [6]]
  - die Entwicklungszeit zwischen 3 und 100 Monaten
  - der Aufwand zwischen 8 und 2000 Mann-Monaten
  - die Zahl der gefundenen Fehler zwischen 10 und 4.000
- Möglicher Erkenntnisgewinn durch statistische Analysen der Variation begrenzt
- Bez. Produkthanforderungen: *Aufwand in MM pro Function Point* wäre eine probate Metrik, um Prozessverbesserungen normiert und anwendungsneutral messbar zu machen [7]

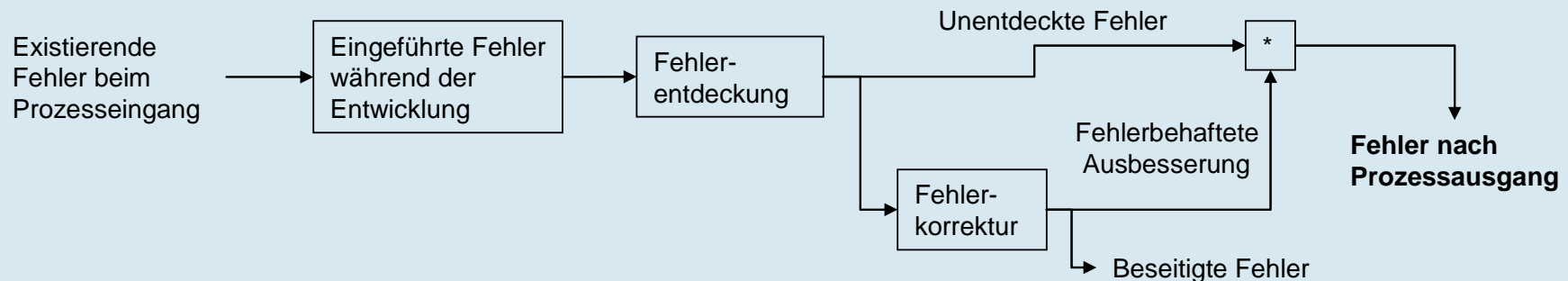


# Metriken und datenbasierte Entscheidungsfindung

- Alle Maßnahmen in einem Six Sigma Projekt sollen ausschließlich auf Basis von quantitativen Daten festgelegt werden
- Zentrale Metrik bei Six Sigma ist die *rolled throughput yield* („=,„ *Sigma-Wert eines Prozesses* „=,„ *defects per million opportunities*) – die Wahrscheinlichkeit, dass eine Produkteinheit den gesamten Produktionsprozess fehlerfrei passiert [HS00]
- Da in der Softwareentwicklung nicht nur Fehlerentstehung, sondern auch Fehlerbeseitigung in allen Phasen stattfindet, kann die *rtv* nicht direkt übernommen werden

→ *defect removal effectiveness* [8]:

$$\text{defect removal effectiveness} = \frac{\text{Beseitigte Fehler in der Phase n}}{\text{Existierende Fehler am Anfang der Phase n} + \text{Eingeführte Fehler in der Phase n}}$$



→ Operationalisierung: Matrix, in der Fehler der entsprechenden Phase, in der sie gefunden wurden, sowie der Phase, in der sie ihren Ursprung haben, zugeordnet werden (ex post-Modell)



## Statistische Analysen als Erkenntnisweg

- Vermutete Beziehungen werden bei Six Sigma gegen empirische Daten getestet, bevor sie als wahr akzeptiert werden [9]
- Gleichzeitig mangelnde Datenqualität und -verfügbarkeit in der Softw.-Entw. [10]
  - Ohne Verständnis der Prozesse, der Ursache-Wirkungszusammenhänge und der Einflussfaktoren auf den Prozess ist bei Interpretation von quantitativen Daten Kausalität von bloßer Korrelation nicht zu unterscheiden
  - Da statistische Verfahren bestimmte Anforderungen diesbezüglich stellen, ist bei Six Sigma in der Softw.-Entw. ein Teil der statistischen Methoden nicht einsetzbar
    - V. a. *DoE - Design of Experiments*
      - Exaktes Messen und kontrollierte Laborbedingungen wie bei physischen Produkteigenschaften in der Massenproduktion und anschließendes Optimieren über eine exakt bestimmbare Faktoreinsatzkombination sind in der Software-Entwicklung nicht denkbar
  - Computergestützte Simulation von Experimenten ist in der Industrie weit verbreitet und auch als Erkenntnisweg bei Six Sigma Projekten in der Softwareentwicklung vorstellbar (Virtual Software Engineering Laboratory [11])



## Statistische Prozessregelung

- Ziel: Aus Vergangenheitsdaten zukünftige Variabilität des Prozesses vorherzusagen, um präventiv eingreifen zu können, bevor der Prozess mit hoher Wahrscheinlichkeit Ausschuss produzieren würde
  - Schlussfolgerung: Im Prozessschritt  $n$  von der Qualität der letzten baugleichen  $x-1$ ,  $x-2$ , etc. Zwischenprodukte auf die Qualität des Zwischenproduktes  $x$
  - In der Softwareentwicklung ist diese Schlussfolgerung nicht plausibel
    - Heterogenität der Artefakte (bspw. Modulkomplexität)
    - Nichtbeachtung des bisherigen Entwicklungsverlaufes (Prozessqualität)
  - Plausibel dagegen ist
    - Schlussfolgerung vom bisherigen Entwicklungsverlauf (bspw. Change History, *dre*, etc.) in den Entwicklungsphasen  $n-1$ ,  $n-2$ , etc. des Artefaktes  $x$  auf seine Qualität in der Entwicklungsphase  $n$
    - Wurde bspw. ein komplexes Modul unter hohem Zeitdruck und vielen Änderungsanforderungen, ist es plausibel anzunehmen, dass die Produktqualität niedrig sein wird
- Metriken und Modelle, die sich auf das Lebenszyklusmodell beziehen, bilden besser die Muster und Gesetzmäßigkeiten der Softwareentwicklung ab als auf sequentiellen Daten basierende Modelle [12]
- *reliability (growth) models* → Fehlerdichte in der Phase  $n$  ist hier eine Funktion der Produkteigenschaften, des bisherigen Projektverlaufs und der aktuellen Projektphase [13]



## Literaturverzeichnis

- [1]: [Töpf04, 38; KoMa06, 767; LSZC03; Step03, 28], [2]: [MaKB04, 43, 59], [3]: [KoMa06; HeNo06], [4]: [KoMa06; HaSc00, 274], [5]: [Ba98, 12ff.], [6]: [PuMy97, 30, 35], [7]: [BF04, 78], [8]: [Kan03, 164f.], [9]: [Mast03, 257], [10]: [Ka03, 470], [11]: [MPR05], [12]: [Ka03, 144], [13]: [Ka03, 187ff.]
- [Balz98] *Balzert, Helmut*: Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Band 2. Spektrum Akad. Verl., Heidelberg 1998.
- [BuFa04] *Bundschuh, Manfred; Fabry, Axel*: Aufwandschätzung von IT-Projekten. 2. Aufl., mitp-Verlag, Bonn 2004.
- [HaSc00] *Harry, Mikel; Schroeder, Richard*: Six sigma: the breakthrough management strategy revolutionizing the world's top corporations. Doubleday, New York 2000.
- [HeNo06] *Hendry, Linda; Nonthaleerak, Preeprem*: Six Sigma: literature review and key future research areas. In: International Journal of Six Sigma and Competitive Advantage 2 (2006) 2, S. 105-161.
- [Kan03] *Kan, Stephen H.*: Metrics and Models in Software Quality Engineering. 2. Aufl., Addison-Wesley, Upper Saddle River 2003.
- [KoMa06] *de Koning, Henk; de Mast, Jeroen*: A rational reconstruction of Six-Sigma's breakthrough cookbook. In: International Journal of Quality & Reliability Management 23 (2006) 7, S. 766-787.
- [LSZC03] *Lindermann, K.; Schroeder, R.; Zaheer, S.; Choo, A.*: Six Sigma: A goal-theoretic perspective. In: Journal of Operations Management 21 (2003) 2, S. 193-203.
- [MaKB04] *Magnusson, Kjell; Kroslid, Dag; Bergman, Bo*: Six Sigma umsetzen. 2. Aufl., Hanser, München 2004.
- [Mast03] *de Mast, Jeroen*: Quality Improvement from the Viewpoint of Statistical Method. In: Quality and Reliability Engineering International 19 (2003) 4, S. 255 - 264.
- [MüPR05] *Münch, Jürgen; Pfahl, Dietmar; Rus, Ioana*: Virtual Software Engineering Laboratories in Support of Trade-off Analyses. In: Software Quality Journal 13 (2005) 4, S. 407-428.
- [PuMy97] *Putnam, Lawrence H.; Myers, Ware*: Industrial Strength Software: Effective Management Using Measurement. IEEE Computer Society Press, Washington 1997.
- [Step03] *Stephens, Kenneth S.*: Six-Sigma and Related Studies in the Quality Disciplines: The Best on Quality Book Series: 14. ASQ Quality Press, Milwaukee 2003.
- [Töpf04] *Töpfer, Klaus*: Six Sigma als Projektmanagement für höhere Kundenzufriedenheit und bessere Unternehmensergebnisse. In: *Töpfer, Klaus (Hrsg.): Six Sigma – Konzeption und Erfolgsbeispiele für praktizierte Null-Fehler-Qualität*. 3. Aufl., Springer, Berlin 2004.

