



Taxonomy of Evolution and Dependability

Integration Engineering – SS 2009 – Andreas Landerer



- Informationen über Massimo Felici
- Definition zentraler Begriffe
- Inhalt des Artikels
- Kernaussagen des Artikels
- Kritische Bewertung und Diskussion



- Wissenschaftlicher Mitarbeiter an der School of Informatics der University of Edinburgh
- Schrieb seine Doktorarbeit über das Thema: „Observational Models of Requirements Evolution“
- Hält Vorlesungen im Bereich Software-Engineering / Software-Architektur
- Veröffentlichte bislang mehr als 30 Publikationen



■ Projekte

- Interdisciplinary Design and Evaluation of Dependability (INDEED)
- Interdisciplinary Research Collaboration in Dependability of Computer-Based Systems (DIRC)
- Validating SQUID in Real Environments (VALSE)
- A holistic approach to the dependability analysis and evaluation of control systems involving hardware, software and human resources (OLOS)



- Taxonomie
 - griech. táxis “Ordnung”, nómos “Gesetz”
 - sprachwissenschaftliche Klassifikation aller Gegenstände und Ereignisse in begriffliche Gruppen bzw. in Kategorien

- Taxonomie in der Informationsverarbeitung
 - Klassifikationen, die eine monohierarchische Struktur aufweisen
 - Durch diese Art der Klassifizierung von Wissensbereichen innerhalb einer Hierarchie entsteht so eine einfache Semantik

Quelle: <http://de.wikipedia.org/wiki/Taxonomie>



■ Software-Evolution

- Wiederholte Anpassung eines Systems an veränderte Anforderungen und Umgebungsbedingungen
- Mit ihr ist unvermeidlich ein Qualitätsverlust verbunden
- Unterbleibt sie allerdings, wird das System nutzlos

Vgl. Ludewig, Lichter: Software Engineering, S. 548.

■ Verlässlichkeit (engl. dependability)

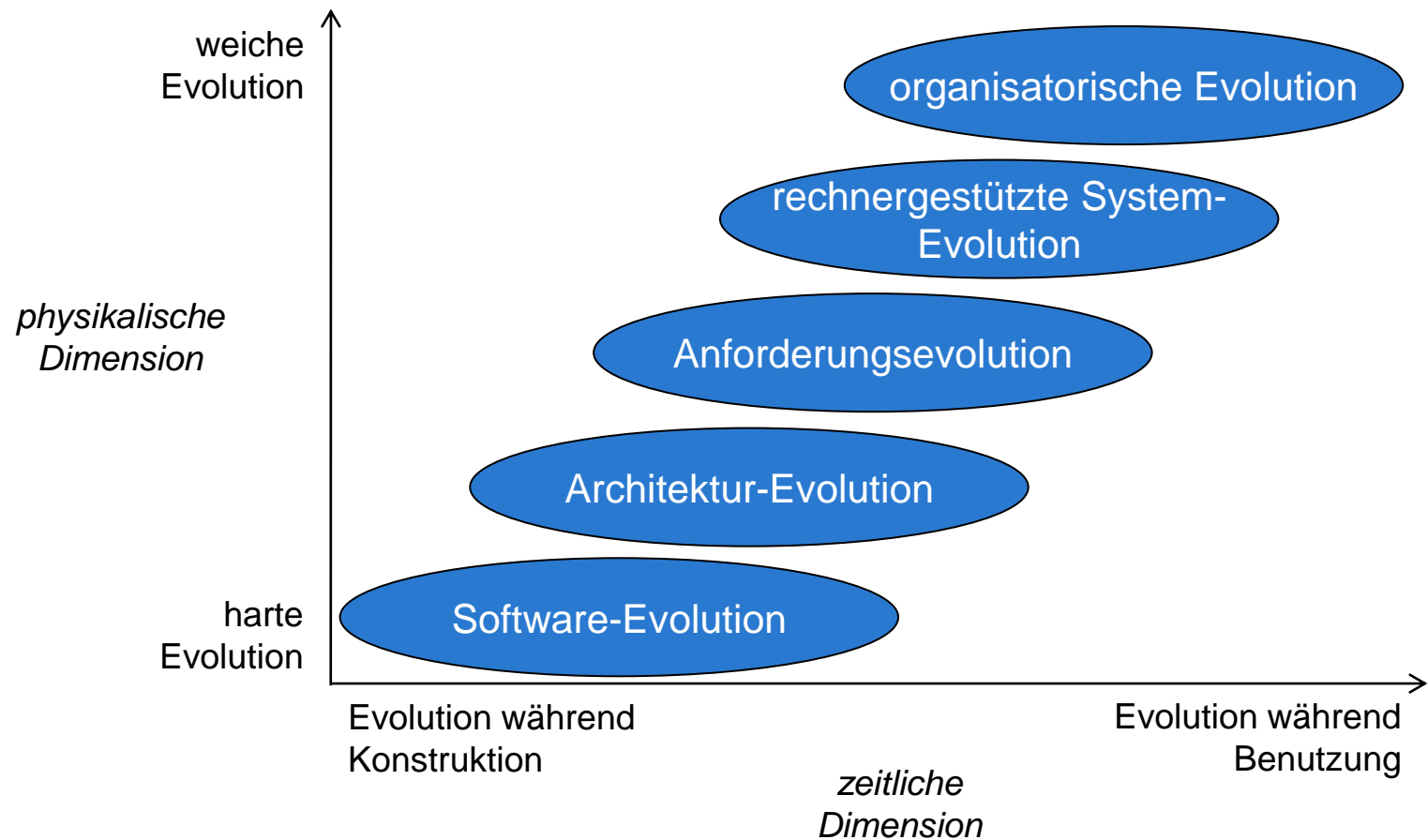
- Übergeordneter Begriff der folgende Fähigkeiten eines Systems beinhaltet:
 - Zuverlässigkeit
 - Robustheit
 - Sicherheit



- 2003 auf dem “Workshop on Unanticipated Software Evolution” (USE) veröffentlicht
- Taxonomie für die Evolution rechnergestützter Systeme
- Zusammenhang zwischen Evolution und Zuverlässigkeit von rechnergestützten Systemen



■ Evolutionary Space





■ Software-Evolution

- Anwendungen unterliegen ständig neuen Benutzeranforderungen und Änderungen ihrer Umwelt
- Prinzipien der Software-Evolution
 - Software-Tektonik: Software muss für zukünftige Änderungen vorbereitet sein
 - Flexibles Fundament: Werkzeuge, Sprachen, Frameworks müssen weiterentwickelt werden
 - Metamorphose: Manipulation des Umfelds, um sich besser an dieses anpassen zu können
- Metriken zur Sicherstellung der Software-Qualität



■ Architektur-Evolution

- Betrachtung der Evolution zur Zeit des Designs
- Architektur sollte stabiler Teil des Systems sein
- Kompromiss zwischen Allgemeingültigkeit und Spezifität

■ Anforderungsevolution

- Analyse stabiler und instabiler Anforderungen
 - Klassifizierung stabiler und instabiler Anforderungen
 - Minimierung der Abhängigkeiten zwischen Anforderungen
 - Identifikation der Art der Änderungen
 - Rückverfolgbarkeit von Änderungen an Anforderungen



- **Rechnergestützte System-Evolution**
 - Rechnergestützte Systeme sind ein Teil von soziotechnischen Systemen
 - Beziehung zwischen technischer und sozialer Evolution ist noch ein aktuelles Forschungsgebiet

- **Organisatorische Evolution**
 - Evolution soziotechnischer Systeme impliziert ebenfalls eine organisatorische Weiterentwicklung und vice versa



■ Zusammenhang zwischen Evolution und Verlässlichkeit

Evolution	Verlässlichkeit
Software-Evolution	Verbesserung durch Fehlerbeseitigung bzw. durch Realisierung neuer Anforderungen
Architektur-Evolution	Kann Verlässlichkeit reduzieren, wenn die Architektur nicht vollständig verstanden wird
Anforderungsevolution	Unzureichend spezifizierte Anforderungen können Verlässlichkeit vermindern
Rechnergestützte System-Evolution	Führt zu Verminderung, da Änderungen an bestimmten Ressourcen (z.B. Software) auch andere Ressourcen (z.B. Hardware) betreffen
Organisatorische Evolution	Unzureichende Koordination von System- und organisatorischer Evolution kann Verlässlichkeit vermindern



- Evolution von rechnergestützten Systemen ist notwendig und muss mit berücksichtigt werden
- Entwicklung einer Taxonomie um die Evolution zu analysieren und zu verstehen
- Evolution hat einen direkten Einfluss auf die Verlässlichkeit eines Systems
- Ableitung eines Frameworks, das evolutionären Phänomene klassifiziert, Verbindungen zwischen ihnen darstellt und sie in Verbindung zu Verlässlichkeit setzt



- Fehlende Abgrenzung des Begriffs Evolution zu ähnlichen Begriffen wie z.B. Wartung, Reengineering, etc.

- Abgrenzung und Zusammenspiel der einzelnen Evolutionsphänomene nur vage beschrieben



- Zusammenhang zwischen Evolution und Verlässlichkeit teilweise einseitig

Softwarequalität wird durch Unzulänglichkeiten in der Wartung schlechter
(z.B. oberflächliche Analyse, zunehmende Redundanz im Code, mangelnde Dokumentation)

