

„Types of software evolution and software maintenance“

Paper review from the “Journal of Software Maintenance and Evolution”

Agenda

- Publikation und Autoren
- Definitionen
- Historischer Überblick
- Motivation
- Ziele
- Klassifikation
- Bewertung

Publikation und Autoren

■ Publikation

- Journal of Software Maintenance and Evolution
- 2001 veröffentlicht, 30 Seiten
- Letzte Zitierung 2008

■ Autoren

- Ned Chapin
- Joanne E. Hale
- Khaled Md. Khan
- Juan F. Ramil
- Wui-Gee Tan



Definitionen

- Evolution

- Genutzt seit den 1960ern
- Keine eindeutige Definition gegeben
- Häufig als Substitut für Wartung gebraucht

- Wartung

- Genutzt seit den 1960ern
- Charakterisiert absichtliche Modifikationen einer Software
- Häufig substituiert durch andere Begriffe

Historischer Überblick I – Ausgangslage

- 1972
 - D. Canning stellt den Artikel “That Maintenance ‘Iceberg’” vor.
- 1976
 - E. B. Swanson stellt eine erste Typologisierung von Softwarewartung vor.
- 1980
 - Lientz und Swanson weisen empirisch nach, dass Softwarewartung hauptsächlich aus kontinuierlicher Weiterentwicklung besteht.
- 1993
 - Der IEEE 1219-1993 Standard wird veröffentlicht
- 1995
 - Der ISO/IEC 12207 Standard wird veröffentlicht
- ...

Historischer Überblick II – Ergebnis

- Vorhandene Typologisierungen basierend auf ...
 - Aktivitäten
 - Intentionen... sind unzureichend.
- Unterschiedliche Definition der Begrifflichkeiten
 - Selbst IEEE-Veröffentlichungen widersprechen sich
- Als Resultat ergibt sich ...
 - Typologisierung basierend auf Nachweisen

- Feinere Granularität der Klassifikation für:
 - Anwender
 - Begrifflichkeiten zu allumfassend
 - Bewertungsschwierigkeiten der eigenen Arbeit
 - Manager
 - Begrifflichkeiten nicht konkret genug
 - Schwierigkeiten in der Ressourcenzuordnung
 - Forscher
 - Begriffsbedeutung und Terminologie unterschiedlich
 - Schwierigkeiten im bei weiterführender Forschung

- Objektiv bestimmbare Klassifikation
 - Basierend auf Beobachtungen und/ oder Vergleichen
- Realistische Wiedergabe der Gegebenheiten
 - Mit Schwerpunkt auf Praxistauglichkeit
- Klare Abgrenzung
 - Software-Evolution und Wartung fokussierend
- Kommunikation erleichtern
 - Unterstützung der vertikalen Kommunikation
- Konsistenzwahrung
 - Gesuchte Übereinstimmung mit vorhergegangenen Arbeiten

Klassifikation I

- Aktivitäten und Prozesse betreffend ...
 - Der Software
 - Des Quellcodes
 - Der Funktionalität aus Nutzersicht
- Beweise dienen als Kriterien
 - Beobachtungen und Vergleiche liefern Beweise
- Kriterien sind unabhängig von subjektiven Einflüssen
- Einteilung von Typen in bestimmte Gruppen
 - Orientierung am Einfluss auf Software und Geschäftsprozess

Klassifikation II

- Zuordnung durch Aktivitäten und Prozesse
 - Der Software
 - Des Quellcodes
 - Der Funktionalität aus Nutzersicht
- Steigender Einfluss auf Software und Geschäftsprozess
- Entscheidungen nach ...
 - Management
 - Qualität
 - ... ausgeklammert.

■ Einfluss auf andere Publikationen

- „For a recent discussion on the topic of classification of activity into types see Chapin et al.” (Lehman/ Ramil, 2001)
- “... only for maintenance projects adding to or modifying the functionality of an existing software application – i.e. enhance maintenance.” (Abran/Primera/Primera, 2002)
- „This typology was further refined by Chapin (...). This earlier work is important and relevant (...).“ (Buckley et al., 2003)
- „Categories along the why axis include: Groomative, Preventive, (...).“ (Jones/Goel/Rugaber, 2007)
- “It could also be used to test and experiment with different types of software evolution and maintenance models.” (Stopford/Counsell, 2008)

Bewertung II

- Artikel
 - Eng an vorherige Artikel angelehnt (Chapin, 2000)
 - Sichtbarer Einfluss auf nachfolgende Arbeiten
- Ziele
 - Objektiv bestimmbare Klassifikation
 - Realistische Wiedergabe der Gegebenheiten
 - Klare Abgrenzung
 - Kommunikation erleichtern
 - Konsistenzwahrung
- Framework
 - Einfach zu erlernen
 - Gute Dokumentation

Bewertung III

▪ Autorenmeinung, Juan F. Ramil

- How do you think your article, especially your proposal on the evidence-based typology, has influenced the newer research to this topic?
 - “(...) It seems that most of the references have been as the Chapin et al paper as a background reading or context. There could be several reasons. Chapin et al may be still the most authoritative view of the problem, or it is difficult to do further research in this topic, or perhaps, lack of interest. (...)”
- Do you think your motivations for the group of practitioners, managers and researchers (p. 4) are today completely fulfilled?
 - “The motivations for each of these groups seem to hold today. (...)”
- Assuming that you could add a new chapter or some new comments to your article, what would be the main points?
 - “Perhaps the empirical validation could be improved. (...)”

Quellen I

- Abran A, Silva I, Primera L (2002) “Field studies using functional size measurement in building estimation models for software maintenance” in Journal of Software Maintenance and Evolution: Research and Practice. 2002, Volume 14, pp. 31-64.
- Buckley J, Mens T, Zenger M, Rashid A, Kniesel G (2005) “Towards a taxonomy of software change” in Journal of Software Maintenance and Evolution: Research and Practice. 2005, Volume 17, pp. 309 – 332.
- Chapin N (2000) "Software Maintenance Types - A Fresh View“ in 16th IEEE International Conference on Software Maintenance (ICSM'00). 2000, pp. 247-252.
- Chapin N, Hale JE, Khan KM, Ramil JF, Tan W-G (2001) „Types of software evolution and maintenance“ in Journal of Software Maintenance and Evolution: Research and Practice. 2001, Volume 13, pp. 3-30.

- Jones J, Goel A, Rugaber S (2007) „Automating Software Evolution“ in Science of Design Symposium. 2007, pp. 14-16.
- Lehman MM, Ramil JF (2001) „Rules for Software Evolution Planning and Management“ in Annals of Software Engineering. 2001, Volume 11, pp. 15-44.
- Stopford B, Counsell S (2008) „A Framework for the Simulation of Structural Software Evolution“ in Transactions on Modeling and Computer Simulation. 2008, Volume 18, pp. 1-36.

Table 1. Approximate correspondence between definitions of types. Correspondence with the Chapin *et al.* definitions exists only when objective evidence confirms the activity. NI—not included; (All)—implicit in all included activities.

Evidence-based (Chapin <i>et al.</i>)		Intention-based definitions			Activity-based definitions		
Cluster	Type	Swanson [8]	IEEE [14,23]	ISO/IEC 14764 [27]	Kitchenham <i>et al.</i> [3]	ESF/EPSOM [33]	
Support interface	Training	NI	NI	(All)	NI	User support	
	Consultative	NI	NI	(All)	NI	User support	
	Evaluative*	NI	NI	(All)	(All)	(All)	
Documentation	Reformative	Perfective	Perfective	Perfective	(All)	Preventive	
	Update*	Perfective	Perfective	Perfective	(All)	Preventive	
Software properties	Groomative	Perfective	Perfective	Perfective enhancement	Enhancements	Perfective	
	Preventive	Perfective	Perfective, or Preventive	Preventive, or Perfective enhancement	Preventive [†]	Anticipative or Preventive	
	Performance	Perfective	Perfective	Perfective enhancement	Corrections, or Implementation change	Anticipative or Perfective	
Business rules	Adaptive*	Adaptive	Adaptive	Perfective enhancement	Changed existing requirements [†]	Anticipative or Adaptive	
	Reductive	Perfective	Perfective	Perfective enhancement	Changed existing requirements [†]	Evolutive	
	Corrective	Corrective	Corrective	Corrective	Corrective	Corrective	
	Enhancive*	Perfective	Perfective	Perfective enhancement	New requirements [†]	Evolutive	

*Default type within the cluster.

[†]A subtype of Enhancements.

Impact of the types of software evolution and maintenance

Impact on software	Impact on business processes Low<----->High	Cluster and type
<p>Low ^</p> <p>-----</p> <p>v</p> <p>High</p>	<p>□□□□□</p> <p>□□□□</p> <p>□□</p> <p>□□</p> <p>□□</p> <p>□□</p> <p>□□□</p> <p>□□□</p> <p>□□</p> <p>□</p> <p>□□□</p> <p>□□□□□□</p>	<p>Support interface</p> <p>Training</p> <p>Consultive</p> <p>Evaluative</p> <p>Documentation</p> <p>Reformative</p> <p>Update</p> <p>Software properties</p> <p>Groomative</p> <p>Preventive</p> <p>Performance</p> <p>Adaptive</p> <p>Business rules</p> <p>Reductive</p> <p>Corrective</p> <p>Enhancive</p>

Swanson (Swanson, 1976) - Intention based classification

- „*Perfective maintenance*“
 - Verbessern der Performanz, Verarbeitungseffizienz oder Wartbarkeit
- „*Adaptive maintenance*“
 - Anpassung des Systems in Bezug auf die Daten- oder Verarbeitungsumgebung
- „*Corrective maintenance*“
 - Korrektur von Fehlern

Software evolution and maintenance - Type classification

Type		Class	
Cluster	Specific type	E	M
Business rules	Enhancive	E	M
	Corrective		
	Reductive		
Software properties	Adaptive	E	M
	Performance		
	Preventive	M	
	Groomative		
Documentation	Updative	M	
	Reformative		
Support interface	Evaluative	M	
	Consultive		
	Training		

TYPES of SOFTWARE EVOLUTION and SOFTWARE MAINTENANCE

NOTES

Tree is read from left to right, bottom to top.
Italics show the type name when the type decision at the left of it is "Yes".
 For "cc", read "change to code".
 * indicates the default type in the cluster.

