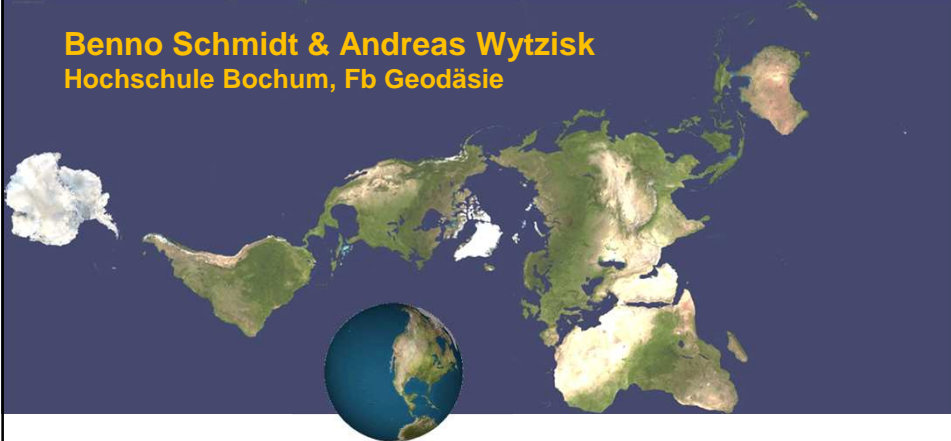


Software Engineering und Integrierte Nachhaltigkeit

Benno Schmidt & Andreas Wytzisk
Hochschule Bochum, Fb Geodäsie



BO 2. Workshop UINW, Stuttgart Sept. 2014

1

Vorbemerkung / Projekt-Hintergrund

- Schwerpunkt "Nachhaltige Entwicklung" an der Hochschule Bochum
 - ❖ ... in Lehre und Forschung
 - ❖ Wir möchten unsere Hochschule in allen Bereichen (CO₂, Baustoffe, Außenbereiche, Mensa, ...) nachhaltig gestalten.
 - ❖ Wir bringen insbesondere Software-Entwicklungen in Projekte ein.



→ *Was ist nachhaltige(s) Software (Engineering)?*

(Diese Frage sollen wir an unserer Hochschule mittelfristig beantworten können.) (Zum Glück gibt es Vorarbeiten...)

BO 2. Workshop UINW, Stuttgart Sept. 2014

2

Erste Meinungen...

- Nachhaltige Software-Entwicklung = "Wir legen Wert auf die Entwicklung **langlebiger** Software-Komponenten"
 - ❖ Vorsehen von Schnittstellen / Abstraktion von konkreter Implementierung, Kapselung in mehrfach verwendbaren Diensten, ...
- Nachhaltigkeit = **Digitale** Nachhaltigkeit
 - ❖ Quellcode-Offenheit, Daten als "unbegrenzte Ressource", ...
- "Informationstechnik ist per se unnachhaltig, da Ressourcen (Materie und Energie) irreversibel verbraucht werden."
 - ❖ Das **Internet benötigt viel Strom**. Die **Herstellung** der Geräte benötigt **Energie, Rohstoffe und Arbeitskraft** (bei **schlechten Arbeitsbedingungen**) und führt zu viel **Elektroschrott**...



2. Workshop UINW, Stuttgart Sept. 2014

3

Ausgangspunkt

- Was verstehen wir im Projekt unter "Nachhaltigkeit"?
 - ❖ Es werden vielfältige Definitionen verwendet.
- Der Ansatz der **Integrativen Nachhaltigkeit** geht über klassische BRUNDTLAND-Definition und Drei-Säulen-Modell hinaus (Ergänzung um sozialwissenschaftliche Aspekte).
 - ❖ BRUNDTLANDT: "... die Bedürfnisse der Gegenwart befriedigen, ohne zu riskieren, dass künftige Generationen ihre eigenen Bedürfnisse nicht befriedigen können."
 - ❖ Drei-Säulen-Modell
 - ökologische, ökonomische und soziale Dimension



2. Workshop UINW, Stuttgart Sept. 2014

4

Integrative Nachhaltigkeit

- Ansatz der Integrativen Nachhaltigkeit (SCHWEIZER-RIES 2013):
 - ❖ Verfolgung der *Effizienz*-Strategie → Aufwandsminimierung
 - ❖ *Suffizienz*-Strategie → nur das wirklich Benötigte brauchen
 - ❖ *Konsistenz*-Strategie
 - Kreislaufführung der eingesetzten Stoffe/Energie
 - ❖ Gedanke der "*fully-functioning society*":
 - Menschen sind in der Lage, sich als Individuen zugehörig zu einer großen Weltgemeinschaft zu empfinden.
 - ❖ *Naturerhalt* und *soziale Gerechtigkeit* als wesentliche Aspekte nachhaltiger Entwicklung.



2. Workshop UINW, Stuttgart Sept. 2014

5

Abstraktes Prozess-Modell (1/2)

- Naive, abstrakte Sicht auf unser Tun als Software-EntwicklerInnen:
 - ❖ Die Entwicklung und anschließende Nutzung von Software lässt sich als *Transformation* von (individuellem) menschlichen Wissen w und Daten d in neues Wissen w_t und weitere Daten d_t (u. a. Software) betrachten.
 - ❖ Zuzuführen sind energetische und materielle Ressourcen r , die teilweise irreversibel umgesetzt werden (Verlust ϵ).
 - ❖ Software vermag im Weiteren als Mittel menschlichen Handeln zu fungieren oder Änderungen im Nachhaltigkeitsbewusstsein bewirken (ψ -Komponente).

$$w, d, \psi \rightarrow w_t, d_t, \psi_t$$

$$r \rightarrow r_t + \epsilon$$



2. Workshop UINW, Stuttgart Sept. 2014

6

Abstraktes Prozess-Modell (2/2)

- Bemerkungen:

- ❖ Der Wissensgewinn Δ_w muss durch Menschen nutzbar bleiben!
- ❖ Der Wissens- und Bewusstseinsgewinn sollte also mit Blick auf die Nachhaltigkeitsziele mindestens so groß sein wie der irreversibel transformierte Anteil der eingesetzten materiellen Ressourcen.

$$\beta(\Delta_w) + \beta(\Delta\psi) - \beta(\epsilon) \geq 0$$

→ quantitative Bewertung ist (ebenso wie Validierung des Modells) äußerst schwierig!

Aber: Grundsätzlich lassen sich Strategien zur nachhaltigen Software-Entwicklung ableiten.

Ableitbare Strategien

- S1** *Gewonnenes Wissen* mit Blick auf die Ziele nachhaltiger Entwicklung *maximieren* (Δ_w).
- S2** Die (irreversibel) eingesetzten *Energiemengen und Ressourcen minimieren* (minimales ϵ , Effizienzaspekt).
- S2'** *Regenerative Energiequellen* für Software-Entwicklung und -Betrieb nutzen (Energie-Anteil in $\epsilon \rightarrow 0$).
- S3** *Langlebige Software-Bausteine* schaffen (Reduzierung von ϵ durch längere Produkt-Lebenszyklen).
- S4** *Breiten Zugang* zu den geschaffenen Software-Artefakten und dem generieren Wissen gewährleisten (Δ_w ; Effiz.; soz. Gerecht.)
- S5** Neue Entfaltungspotenziale des Integrativen Nachhaltigkeitsbewusstseins schaffen ($\Delta\psi$).

Konkrete Maßnahmen (1/3)

- Unnötige Mobilität der Projektbeteiligten vermeiden (!)
(Energieeffizienz)
- Während der Anforderungsanalyse weitreichende Partizipationsmöglichkeiten für alle Stakeholder gewährleisten
- Langlebige Interfaces anbieten (Anwendungen bleiben bei Änderung der Implementierung lauffähig)
- Software-Schnittstellen nicht unnötig komplex entwerfen
(Beispiele: OpenLS, OGC-SOS, ...)
- Millionenfach aufgerufene Server-seitige Funktionen algorithmisch optimieren (CPU-Last / Energieverbrauch)

Konkrete Maßnahmen (2/3)

- Breite Software-Nutzung ermöglichen (Benutzerschnittstellen für technikferne Zielgruppen, ältere Menschen, barrierefrei, in anderen Kulturkreisen)
- Keine Komponenten/Funktionen verwenden, die unangemessen hohen Hardware-Voraussetzungen stellen (!)
(Negativ-Beispiel: Office-Anwendungen; nicht zu Kauf eines neuen Computers zwingen)
- Offline-Betrieb von Web-Anwendungen gewährleisten und Caching-Mechanismen unterstützen
(Negativ-Beispiele: Web-Mapping auf mobilen Endgeräten, YouTube, ...)
- Auf Lauffähigkeit der Web-Anwendung bei niedrigen Bandbreiten achten (insb. auf mobilen Endgeräten hier wie in strukturschwachen Regionen dieser Welt)

Konkrete Maßnahmen (3/3)

- ... und natürlich die Bereitstellung von Anwendungen, die mit Blick auf die Nachhaltigkeitsziele eine sinnvolle Funktion erfüllen:
 - Papiereinsparung durch digitales Büro
 - Online-Banking statt Fahrt zur Bank
 - Car-Sharing-Apps / Mitfahrzentralen
 - GIS-Werkzeuge zur Optimierung des Standorts einer WKA
 - Anwendungen, welche die Partizipation an NH-relevanten Fragestellungen ermöglichen (im Sinne einer Citizen-Science)
 - ...
 - Aufbau einer Web-Community, die Mitglieder zum Sparen von Energie und Ressourcen motiviert, so dass Bilanzgleichung positiv.



Untersuchungswerte Maßnahmen

- Wie viel Energie wird für Server-seitige Komponenten tatsächlich benötigt?
- Wie nachhaltig ist Open-Source? (Nachbauen existierender Lösungen durch aufwändig zu koordinierende Community?)
- Welchen Stellenwert hat die Kommunikation wesentlicher konzeptueller Elemente des Software-Entwurfs (Wissenstransfer statt "tacit knowledge") in Bezug auf die Ziele nachhaltiger Entwicklung?
- Offene Software-Schnittstellen (u. U. Blackbox-Implementierungen) als wesentlicher Beitrag?



Fazit (1/2)

- *(Integrativ-) Nachhaltige Software zeichnet sich dadurch aus, dass die direkten und indirekten negativen Auswirkungen auf Mensch, Gesellschaft und Umwelt, die sich aus der Entwicklung, dem Betrieb und der Verwendung der Software ergeben, minimal sind. Zudem sollen sich mit Blick auf eine (integrativ-) nachhaltige Entwicklung durch die Software langfristig positive Auswirkungen ergeben.*
 - Präzisierung der Definition von DICK, NAUMANN et al.:
(Zeitvariante) Gesamtbilanz muss positiv sein.



Fazit (2/2)

- Effiziente, langlebige und frei zugängliche Software ist nicht per se (integrativ-) nachhaltig.
 - Aspekte der Konsistenz und sozialen Kontexte
- Dennoch gibt es zahlreiche Empfehlungen im Sinne selbstverpflichtender "Gesten".
 - **Erweiterung der Liste (mit Hinweisen zur Bewertung)?**
- Sensibilisierung der Akteure (Entwickler, Projektleiter, IT-Leiter, ...) für das Thema ist wichtig.
- Weitere Information:
 - ❖ Tagungsband und interner Bericht
 - *Anregungen / Ergänzungen sind höchstwillkommen!*



Einige weitere Forschungsfragen (1/2)

- Wie lässt sich die Nachhaltigkeit von Software und Software-Entwicklungsprozessen praktikabel bewerten?
- Welche konkreten *praktischen Hilfestellungen* lassen sich Software-EntwicklerInnen geben? (Wo lässt sich etwas bewirken?)
 - *Hier hätten wir gerne so etwas wie eine Checkliste oder einen Leitfaden / in weiter Ferne: Alternative "Patterns" oder Warnungen in IDE ...*
- Beobachtung: Zugewinne an Performanz, Speicher etc. werden sogleich ausgenutzt (*Rebound-Effekt*). Welche Erkenntnisse/Lehren/Konsequenzen lassen sich hieraus ziehen?

Einige weitere Forschungsfragen (2/2)

- Wie lässt sich die Kluft zwischen Software- und Wissensproduktion und gesellschaftlichem Handeln in Richtung nachhaltiger Lebensweisen überwinden?
- Welche Anforderungen ergeben sich (über den Software-Aspekt hinaus) für eine nachhaltige (Geo-) Fachdatenerfassung, -haltung und -bereitstellung?

Die letzte Folie...

- Wir suchen Kooperationspartner im Umfeld des vorgestellten Arbeitsfeldes!
- *Danke für Ihre Aufmerksamkeit!*

Urhebernachweise zu Buckminster-
Fullers Dymaxion-Projektion:
NASA
CC-BY SA 3.0

