Notes for Seminar / Master Thesis Literature

## Sustainability in Software Engineering – A SLR (Penzenstadler et al.) 2012

* FIRST systematic literature review -> provide body of knowledge
* Not yet: overview about the current state of art (Sust. In SE)
* Definition Sustainability:
  + Brundtland report – sustainable development, 3 dimensions Society, Economy, Environment
  + + 4th dimension: human sustainability (relevant according to R. Goodland – [6])
* Sustainability ASPECTS in Software Engineering 🡪 möglichkeit für strukturierung für mich?!
  + Development process viewpoint
    - Development process
      * Initial dev process -> responsible use of human resources WHILE developing
      * Initial conceptual & constructional developement
    - Maintenance process
      * Sustainability of system during maintenance – or OF maintenance process?! -> rather the latter
      * Aspects like continuous monitoring, Knowledge mgmt (+ bug fixing etc?)
  + Product viewpoint
    - System production
      * Sustainability of the product with regards to RRSOURCES used for production – sustainability of used hw products etc?
      * Aspects like mass production, logistics, etc. (so non-dev stuff)
    - System usage 🡪 hier wäre dann SECOMO zu finden?!
      * Sustainability during usage of the system as a product
      * Aspects TRIGGERED by the software, e.g. impact on environment (🡪 eco-costs) or using green business processes, (probably also – social sustainability!)
* From first to last -> growing impact 🡪 GROWING IMPROVEMENT POTENTIAL!

**Results**

* RQ1: 2012 – how much activity in the last 20 years? 🡪 increase in the last 2 years 🡪 so around 2010 it started to come into focus!! -> nice for introduction or Sust. SE chapter!?
* RQ2: Adressed Research Topics – general: *“The dimensions of the taxonomy are the degree of domain specifity, from general purpose to domain-specific research and the indexing between analytical approaches (frameworks and assessment) and constructive approaches (methods and tools). The taxonomy shows a tendency towards domain-specific, constructive approaches. There are not many publications rated as general purpose, and there is little methodical guidance for supporting sustainability.”* 🡪 most research goes into the direction of domain-specific, “constructive” approaches, but usually for a certain specific domain -> general method missing 🡪 Sust. SE Chapter? General part
  + Taxonomy see Figure 2 (…)
* RQ3: Limitations
  + High complexity (due to high connectivity of all the sustainability aspects) -> clear concepts and models needed
  + High domain-specificy 🡪 so far only very specific frameworks/methods exist (as effective approaches might really need specific domain knowledge?!)
  + (software engineering) 🡪 so far only one approach 🡪 *“An encompassing reference framework for SE is still missing” p. 6 / 37* -> from 2012 -> THEN GREENSOFT came, BUT … and then explain what is still missing for greensoft? – BUT greensoft was already part of the SLR ?!!?
* RQ4: How is Sustainability Support performed?
  + (most =) CONSTRUCTIVE support (so what you can actually do) 🡪 use this as structure?
    - Frameworks
    - Models
    - Methods
    - Metrics
    - (Tools / Prototypes – e.g. greentracker / website measure trier?)
    - …but usually this is very DOMAIN SPECIFIC, e.g. the metrics or the frameworks (p. 6 / 7) 🡪 would be nice to have something more generic 🡪 SECOMO – but ! it is also rather for (object) oriented software systems!! -> limitation!
  + + what about guidelines / manifestos?!
* RQ5: Which methods are IN USE? ~~-> what about the practical side of it all???~~
  + Which TECHNIQUES are used in the field?
    - Traditional SE techniques
    - Methods from other disciplines
    - E.g. ER-Modeling, Neural Networks, Cost Calculations, Life-Cycle Analysis
* RQ6: Case Studies available – so when things have been used & studied in practice??
  + A few – but rather small ILLUSTRATIVE case studies to propose a method
  + 🡪 THIS IS MISSING -> something to understand “HOW TO DEVELOP SOFTWARE FOR SUSTAINABLE SYSTEMS” (p. 7) or sustainable software or or or
* RQ7: Which domains are already considered?
  + Systems & Knowledge (analytical, general purpose)
  + Technologies & Methods (constructive, general purpose)
  + Education (somewhere middle)
  + Disciplines (domain-specific, analytical(?))
  + Application & Implementation (domain-specific, constructive)
  + SECOMO would be (Constructive, general purpose) ???

Discussion 🡪 NOT AS MUC AS EXPECTED FOR SUSTAINABILITY IN SOFTWARE ENGINEERING ITSELF, more around …. Concepts, concrete solutions…

## Systematic Mapping Study – SE4S - Penzenstadler et al. (2014)

* Provide overview of Status of research in the SE4S field!
* Extend first study from 2012 🡪 then not much on actual results in SE field -> but much work in between those studies -> now more results!

Introduction about Sustainability / Motivation

* Definition of Sustainable Software
  + The term Sustainable Software can be interpreted in two ways: (1) the software code being sustainable, agnostic of purpose, or (2) the software purpose being to support sustainability goals, i.e. improving the sustainability of humankind on our planet. (p. 1)
  + sustainable software is energy-efficient, minimizes the environmental impact of the processes it supports, and has a positive impact on social and/or economic sustainability (p. 1)
  + These impacts can occur direct (energy), indirect (mitigated by service) or as rebound effect (p. 1) -> (Hilty et al – The relevance of ICT… - 2006)

Study Design – Research Questions !!

* RQ1 What research topics are being addressed?
* RQ2 How have these research topics evolved over time?
* RQ3 How is sustainability support performed (e.g., models and methods)?
* RQ4 Which of those models and methods are used in practice?
* RQ5 Which research type facets have been considered in the contributions?
* RQ6 Which application domains have been considered?
* RQ7 Which research groups are most active and what is the distribution between academics and practitioners?

Results

* RQ1 -> what topics? -> Topic Cluster Model (referring to abstracts)
  + Future of Society, Urban Architecture and integration, ENERGY EFFICIENCY, LIFE CYCLE ASSESSMENT; ENVIRONMENTAL MANAGEMENT, Smart Grids, CLOUD services, CARBON CONSUMPTION, Traffic strategies, VIRTUALIZATION (???)
  + Popular research areas: Software Engineering Process (topic 10) & Software Desing & Software Quality (topic 5)
  + Majority of publications are in;: Software Design, Engineering Management, Models and Methods, Process, Quality, Requirements
* RQ2 -> how has it evolved? 🡪 last two years 40 NEW RELEVANT PUBLICATIONS 🡪 an increase in the last few years!!! 🡪 sth. For introduction?? In terms of comparing the 12 to the 14 study  
  🡪 strong development over the last four years (-> since 2010)
* RQ3 -> what methods / models are USED in the publications????? ☹ ☹
  + Mostly: Software engineering methods / tools
  + + energy management: measuring devices!
  + But also: general purpose methods (interviews, surveys) ….
  + 🡪 variety of models and methods
* RQ4 -> NOT MUC in EVALUATION or EXPERIENCE 🡪 use this for the PRACTICE part!!!!!!!
  + Question: “is not really triggering a state of practice at all or whether it is simply not published much on yet” ?
  + 🡪 combine with “ivory tower” paper??
  + Very limited usage in practice in the evidence
* RQ5 -> Research methods?
  + Exploratory or Solution -> so about practical, specific projects
  + Not much about experience / evaluation!! 🡪 more reports on experience necessary – lack in practical evaluation!!
  + 🡪 most prominent research type facets: Exploratory, Solution
* RQ6 -> Application domains? 🡪 mention in some sort of categorization??
  + Software Engineering & lifecycle (rather generic approaches)
  + Energy efficiency
  + Services, Mobile & Cloud (more specific..)
  + Business & Economics (business processes, organizational issues)
  + Systems Engineering & ICT (MORE THAN software -> consider green ICT in general)
  + ULS Green Computing (Large-scale systems!!)
  + (other domains: Mechanics & Manufacturing, Nature & Agriculture, Metropolitan Areas & Housing…)
  + Software Engineering EDUCATION !!! -> important factors
* RQ7 -> most active research groups? :D
  + 80 % of publications coming from academia!! -> another indicator -> practical view is missing!
  + 3 big research groups ->
  + 🡪 3 active big groups, all over the world, unbalanced distribution between academia (80%) and industry

Conclusions

* Over the past few years: SE4S has gained MORE ATTENTION! -> not much report on “establishment in practice”
* Industry “hype” is recognizable (like with Green IT, now: “Sustainability”) -> green IT is now more established inpractice -> hope that this will happen for sustainability, too!
* Still “immature” research area -> not much evaluation / experience papers
* Research community is still “forming”
* But broad range of topics already
* Main results: see above

## Green in Software Engineering – Calero / Piattini, 2015 -> DEFINITIONS

Introduction

* Sustainability is becoming more important – Earth Hour, UN projects, ..
* Sustainability projects
* ICT / technology -> can help to achieve changes, goals of projects, BUT: has also negative impacts – “often responsible for major environmental degradation”  
  🡪 *“This dual aspect of technology means that organisations also face two challenges: they need to have more sustainable processes and they must produce products that contribute to a more sustainable society”* (p. 2) 🡪 use this for role of IT in sustainability
* Some numbers about energy consumption 🡪 Smart2020
* 🡪 NEED to control use of ICT -> to reduce impact -> especially software (here)
* Role of sustainability in BUSINESS -> Strategic Sustainability -> it is important to have it, competitive factor / CSR
  + ISO 26000 Standard for CSR -> be cautious about environment <https://www.iso.org/obp/ui/#iso:std:iso:26000:ed-1:v1:en>
  + Triple Bottom Line – “financial performance, environmental practices and corporate social responsibility (CSR)” p. 3
* WHAT is MISSING? -> new area, important and becoming a necessity -> BUT clear concepts / definitions missing

Sustainability

* Collins dictionary: **maintain** at a steady level w/o exhausting natural resources or causing damage
* (others) using a **resource** so as **not to deplete, exhauste, damage** it and so to ensure its existence for **future generations**
* Brundtland: ‘meet needs of present without compromising the ability of future generations to satisfy their own needs’ -> UN /common future: 3 dimensions, Society, economy, environment 🡪 most widely used!!!
* Interesting definition: (Adams WM (2006) The future of sustainability. Re-thinking environment and development in the twenty-first century: technical report, IUCN)
  + economic development, social development and environmental protection
  + (see p. 4/5)
* Main factors
  + Ability / capacity of sth to endure / last long / be maintained
  + Focus on RESOURCES used (not to exhauste them)
  + In order to ensure a good life for FUTURE GENERATIONS

Information Systems Sustainability

* IS discipline is central in “creating an ecologically sustainable society” -> because of the long time this discipline has already been important
* IS = transformative power in terms of changing PROCESSES towards more sustainability (?) -> create environmentally sustainable organisations
* Most work done so far: “Green IS” ????

ICT / IT Sustainability

* What can be done?
  + Align IT processes with “core principles of sustainability: reduce, reuse and recycle” ???
  + Find ways to USE IT in business processes to deliver sustainability (greening BY it)
  + Other ways: Dematerialisation / Virtualisation
  + Increase efficiency of IT

Software Sustainability

* Most work so far: improving sustainability (eco) of DATA CENTERS – as the total energy consumption is of course higher as the average office
* Most intuitive way to SUSTAINABLE SOFTWARE – improve power consumption (much has been done to hw, but not much to software..)
  + (capra 12) -> energy consumption / efficiency not often focus of SDLC and methods / tools / quality aspects
  + (source?? S. lund paper) Never been a key requirement!! (not part of NFR?!)
  + (also penzenstadler etc.?!)
* Software HAS an influence!! Neg. & pos. (Easterbrook???) -> also part of sust. Se part!
  + DIRECT & INDIRECT IMPACTS (see Greensoft papers)
    - Direct impacts – related to resources & energy consumption during production / use of the software
    - Indirect impacts – related to usage of software indirectly and long-term systemic effects
  + BEHAVIOUR of sofwatre has influence on energy consumption (s. steigerwald, argawal)
  + CODE itself can be sustainable (s. purpose -> ; s. sustainable programming)
  + Processes around can be influencing in positive way
  + DISPOSAL of hw / sw, ?! (s. amsel / argawal?)
  + TODO -> CHECK!
* Definition Sustainable Software
  + Penzenstadler (systematic mapping study) -> vey nice!!
  + Dick / Naumann etc. -> 11-13 papers sustainable dev, sustainable sw, sust. SE) + greensoft and others
* Sustainable Computing -> (Mocigemba, 06)
* ATTENTION – “sustainable” & “green” are sometimes used synonymously, but careful – does not fit with the definition of sustainability!!! -> depends on how you want to understand it
  + Here: this is faulty!
* Important research topic!!
  + Raise awareness with COMPANIES developing software, with USERS, and so on (Admins, etc.)

Software Engineering Sustainability

* Within SE, not so many concept wrt sustainability exist (see SLR by Penzenstadler), a bit more over the past years (2010 – 2015) (penzenstadler mapping study) 🡪 concern & interest for research field is existent
* Stuggle: how to integrate sustainability aspects in SE?? -> similar structure in my paper?! – first why software, then why SE & HOW? -> then present what exists!
  + (Generic model penzenstadler) 5 dimensions
    - Individual sustainability (satisfaction of delveopers/human capital) during SE
    - Social sustainability (effect of software system on society?)
    - Economic sustainability – maintain assets, stakeholders investments – be economically sustainable(?)
    - Environmental sustainability – protecting natural resources, reducing waste etc.
    - Technical sustainability – long-time use of systems!!! -> adaptability of the system for future change!
* **Definition of sustainable SE in Table 1.1**. (p. 10)
  + (Amsel etc. – Towards sust. SE – NIER) 🡪
  + (Manteuffel, Loak..12 – A systematic mapping study)
  + (ISO26000 – Guidance on social responsibility)
  + (Tate, 06 – Sust. Software development)
  + (Dick, Naumann, 2010 - Enhancing SE processes) !!!!
  + (Kern, Dick…13 – Green SW and Green SE)
  + (IDC 09, Aid to recovery)

From Sustainability to Greenability

* Sustainability: 3 dimensions 🡪 in the context of technical aspects -> FOCUS on ENVIRONMENTAL / GREEN Dimension
* GREEN IS & IS Greenability used synonymously (???) -> no, I prefer: Green IT and GREENING by it
* .. what does green mean for each of the levels?

Green IS

* Chen et al. -> Green IT & Green IS target products and practices that aim at aspects like pollution prevention, and other environmental aspects (??)

Green ICT / IT

* Various definitions of the concept
  + Relation of IT & Energy efficiency (Capra 12 – is software green?)
  + Using Technology efficiently + keeping 3BL in mind (Bachour, Chasteen, 2010)
  + Also includes eco-computing & green computing
* Practices cover several phases:
  + Acquisition, recycling, final disposal
* (Murugesan, 11) Green IT is the study and practice of designing, manufacturing, and using computers, servers, monitors, printers, storage devices, and networking and communications systems efficiently and effectively with minimal impact on the environment. It includes environmental sustainability, the economics of energy efficiency, and the total cost of ownership, which incorporates the cost of disposal and recycling. Green IT is also about the application of IT to create energy-efficient, environmentally sustainable business processes and practices -> cite, which page???
* (other definitions -> p. 12 – 13)
* Best definition (due to author) – (penzenstadler, generic model): … (TODO – die raussuchen)
* BUT ISSUE: for a long time, it only focused on HW ☹ (Erdelyi, 2013)

Green by IT vs. Green IN IT -> can be broken down to software!!

* What role does IT play? (see also: Green sw and green SE – Kern, Dick, 13)
  + A) producer of emissions itself 🡪 make IT itself environmentally friendly -> GREEN IT
  + B) Enable to reduce emissions in other areas 🡪 make sth. Environmentally friendly BY IT
* IT = software and hardware -> so the idea can be applied to SW too
* Other concepts: Green IT 1.0 and 2.0
* With focus on waste –
  + Green in IT = producing as little waste as possible in IT lifecycle
  + Green by IT -> produce as little waste as possible BY USE OF IT
* Penzenstadler 13 – Towards: Sust FOR and IN SE … but rather: “Green” for / in SE

Green Software

* Impact of software on energy consumption – during WORKLOAD processing and IDLE
* So far: mostly hw focus… -> focus on software is important, too!
* Common definition of “green software” is missing.. !? (Abenius 09)
* Definitions…

Green BY Software vs. Green IN Software

* Green BY Software – sw that helps the environment & ecologically aspects
  + Software for different domains, with the goal of preservation of environment
  + SW that helps to MANAGE energy consumption
* Green IN Software – have more Environment-friendly software – over the whole lifecycle
  + How to make software more (sustainable?!) green

Green in Software Engineering – focus of the book

* Include “green” practices in the software development process

## What does Sustainability mean IN and FOR Software Engineering? - 2013

(Penzenstadler, Birgit, 2013)  
🡪 Aims at defining sustainability in this context & propose possible actions

* Missing Common Understanding!
  + Traditional SE has not supported sustainability so far!
* Sustainability is not a tangible concept itself, and software engineers are aware of the general idea, but it is not clear what it means in the context of SE
* Problem: No common understanding: in traditional SE methods, sustainability is not supported 🡪 several aspects are tackled (e.g. Green IT, knowledge mgmt.), but NO COMMON view
* Approaches to define sustainability in SE
  + Mahaux et al. ->
  + Naumann et al. ->
  + Question of perspective; here: Sustainability is satisfaction of needs for humanity in ecosystems over multiple generations
  + Penzenstadler suggests: Consider System, Function and Time horizon as dimensions under which to define sustainability (in SE)

Aspects of Sustainability in Software engineering -> like in SLR -> here more details!?

🡪 all four aspects are necessary for an holistic approach to reach sustainability in software engineering, BUT author: system USAGE aspect has biggest impact 🡪 thus here very important to support approaches / methods models 🡪 SECOMO!

* Development Process Aspect -> *“development with responsible use of ecological, human, and financial resources” (p. 3)*
* Maintenance Process Aspect -> *“maintenance period until replacement […] includes continuous monitoring of quality and knowledge management”* (p. 3) -> critical!?
  + Also includes “reengineering and refactoring” to cover the lifecycle phases of replacement and disposal (???)
  + But could be better emphasized
* System Production Aspect -> *“Sustainability of the software system as product with respect to its use of resources FOR PRODUCTION” (p. 3)* -> for example by following Green IT principles, using energy efficient HW components, optimizing logistics etc.
* System Usage Aspect -> sustaintaiblity (impacts) triggered during the USAGE phase -> e.g. *“responsibility for the environmental impact and designing green business processes” (p.3)*  
  🡪 HERE secomo fits in!

Actions to IMPROVE sustainability(?) 🡪 ideas / best practices?!

* Concerning the SOFTWARE DEVELOPMENT **PROCESS**
  + “common artefact model for documentation” -> improve REUSE or REFACTORING
  + Knowledge management -> share knowledge/best practices, avoid double work, improve initial development & maintenance process
  + Continuous quality assessment -> e.g. Albertao;
  + Optimizing RESOURCE usage, e.g. wrt *energy consumption using development equipment* (overlap to System Production Aspect??) or (operational) general office infrastructure (-> meta-ebene? :D) (but this is independent of specific software engineering practices)
  + “working offline” -> improves communication (as people are forced to work directly together) and probably creativity, not so much resources
  + “energy-aware software design”, see Naumann et al () -> sustainable web development?!
  + PROJECT MANAGEMENT assessment regarding sustainability -> “Sustainability Maturity Model” -> analyze & set goals for improvement
    - *G. Silvius and R. Schipper. A Maturity Model for Integrating Sustainability in Projects and Project Management. In 24th World Congress of the International Project Management Association, 2010.*
* Concerning the SOFTWARE SYSTEM as **PRODUCT**
  + Life-Cycle Analysis -> measure sustainability of the system during production and usage, with sustainability metrics
    - E.g. Alberato et al (but rather general software quality metrics, very generic)
    - Bell and Morse (Sustainability Indicators – Measuring the immeasurable?)
    - SECOMO -> estimate -> use this to improve ALREADY during development, not as a second step afterwards!
  + Redesigning / optimizing Business Models / Business processes -> to minimize rebound effects
    - Principles of sufficiency instead of maximizing profits…
    - DO THIS during requirements analysis ->
    - NEED: *“alignment of sustainability principles with business goals in order to derive software system requirements” (p. 4)*
  + ASSESS software of the BUSINESS PROCESSES to measure THEIR sustainability -> e.g. their energy consumption
    - *Patricia Lago et al. The service greenery – integrating sustainability in service oriented software. In International Workshop on Software Research and Climate Change (WSRCC), 2010.*

## Sustainable Approaches and Good Practices in Green Software Engineering - 2012

(Agarwal, Nath, Chowdhury - 2012)

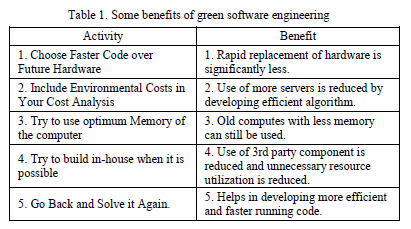
Green computing is good (ethical) AND profitable 🡪 the ultimate solution

* Need for sustainability: ICT causes problems partly…
  + Power consumption goes up (through ICT)
  + pollution increases, waste, …

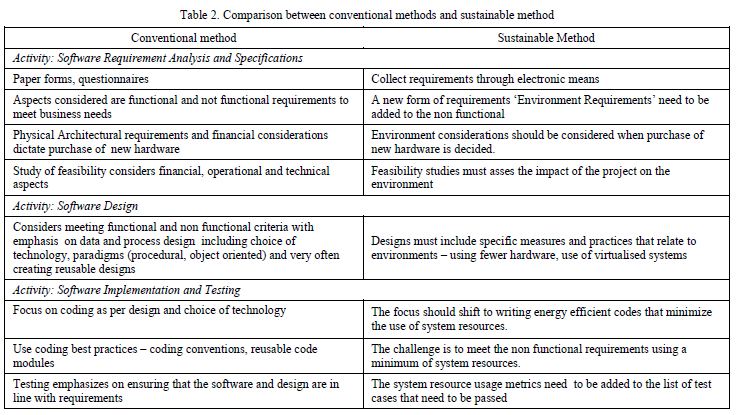
Thus 🡪 needs to solve it, too.

* Green IT = “reduce the consumption of power in H/W design”
* BUT computer software: “lack of models, descriptions, or realizations”, but NECESSARY; “hardly any systematic algorithms available that try to integrate sustainability aspects in software product design and development” 🡪 why software and what is needed?
* NO clear definition of “Sustainable Software” / no “criterions”, not considered in SW development process models

Sustainability Aspects of Software

* Green Computing / Green IT: Design and use of ICT should be environmentally friendly
  + Manufacturing: “reduce the use of hazardous materials, maximize energy efficiency during the products’s lifetime, and promote the recyclability”.
  + “SAVE ENERGY” – “reduce the environmental footprint of technology”
  + Software / Services area: virtualization / capacity management / data-center planning (quotes) 🡪 intelligent solutions contribute to Green IT
  + PROFITABLE?! = reduction of resource consumption = cost savings, too
* NOW: ‘Green Software Engineering’ = try to “apply these ‘green’ principles known from hardware products also on software products, software development processes and their underlying software process models’
  + SDLC traditional focus: “basic [] characteristics like usability and maintainability”
  + “Green” SE should focus on the “ecological, social, human and economic compatibility of a product during its whole lifecycle”
* **SUSTAINABLE SOFTWARE:** 
  + Definition: sw is considered sustainable if “direct direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software are minimal” or even lead to a positive effect 🡪 **geklaut von NAUMANN, Greensoft, 11 !!!!**
  + Direct impacts = energy & resources necessary for “produce, use and dispose” the sw
  + Indirect impact = “effects that result from using the software product” on other aspects of the surroundings
  + BACKGROUND: software performance = basic computing resources (CPU / memory) 🡪 high resource demand leads to 🡪 need for new hardware 🡪 waste ☹ + 🡪 more power consumption
* Idea: Maintenance = “fresh development effort” = “repeat […] the entire SDLC process” = increase carbon footprint 🡪 prefer less maintenance need! / longer lifetime
* Virtualization: (definition?) = initial intention = “cut down costs of […] servers and desktops”, but also led to more sustainable software (…why?)
* **GREEN SOFTWARE ENGINEERING:** Naiive idea: software “is automatically green”, or has no environmental impact (why?? 🡪 cause there is no physical artefact that could e.g. cause pollution?) – but it CAN have an impact, and the existing impacts vary!
  + **ISSUE:** “software developers don’t think in terms of developing green software” – because software is assumed not to have an environmental impact - and that needs to change!   
    Examples for negative results of current practices:
    - Improved hardware compensates the existence of slow / inefficient software, thus nothing is changed about it as nobody notices the performance degradation
    - Cost analysis: focus on man time 🡪 reduce development time! 🡪 instead of creating a better algorithm that might “take less time to execute”
    - PATTERN OF REUSE – YES! It is a good an common thing, but if it is “thousands of times larger than the portions [the developers] actually want to use” – it might have the negative effect on memory / execution time
  + **POSSIBLE SOLUTIONS:** improve software quality, as sw is reused 🡪 time savings will multiply and positive effect on sustainability; don’t always reuse, do in-house dev. If possible etc. ! (see table)  
    

**Improvements in the SDLC models (more greener Aspects)**

* Software Development Life Cycle (SDLC) – meet functional and non-functional requirements (🡪 reference?? Balzert?) – functional = do what It should & non-functional = quality as expected
* Conventional: Sustainability is NOT considered! Environment, carbon footprint
* Comparison of Conventional w/ Sustainable Methods:  
  
  + Requirements Engineering: Include “Standard Set of Environmental Requirements”!! When considering new hardware in architectural decisions, measure impact on environment in feasibility (come up with “quantitative metrics” to use as acceptance criteria); but also simply to transform ways of collecting requirements into more sustainable way = collect requirements electronically
  + Software Design: when designing, environmental measures must be considered (degree of virtualization, hardware requirements, …)
  + Implementation / Testing: write energy efficient code, “environment friendly best coding practices” (how? 🡪 somebody written more about that?), shift from ONLY looking at best practices to how to achieve minimum system resources? (especially: remove any redundancy, 🡪 “exploit parallel, multiprocessor architectures”), test also according to environmental metrics, e.g. resource usage!
  + 🡪 SUPPORT the conventional methods, NOT replace them!
* Further good practices…
  + Focus on reusing infrastructure that already exists instead of getting new hardware 🡪 better use of existing intrastructure!! (see green grid article?)
  + Architectural considerations: use virtualization / cloud computing (-> on-demand use, better load balancing, …)

**Conclusion**

* The actual impacts of sw on the environment are not completely clear – more energy saved than consumed by better systems?
* Anyway: Include Environmental requirements in SDLC practices!
* This can be a competitive advantage for the company AND better for the environment!

## A generic model for sustainability with process and product-specific instances

* 5 dimensions

## Safety, Security, Now Sustainability: The NFR for the 21st Century – Penzenstadler et al. – 2014

Abstract

* Software Engineers can improve sustainability – with Focus on First-, Second- and Thrid-Order impacts
* Last decades – software became ubiquitous -> need for SAFETY and SECURITY became evident -> became non-functional requirements!
  + Other NFR’s were efficiency, reliability, usability
  + Safety and security were added later (ISO/IEC 25010:2011)
* 21st century -> software is central in companies operations / industy
  + Thus play a central role in the unsustainable way our world works
  + Not intentionally, but it happens – everything has an impact on sustainability
  + *“It isn’t civilization’s intention to harm the Earth, but the collective sum of our individual actions, which often favor local convenience over global responsibility, added to the effects of the societal structures we’ve created, lead to negative consequences for our environment” (p.1)*
  + SOFTWARE has the opportunity to make a difference due to its ubiquity, to support society to make a change -🡪 SUPPORT GREENING THROUGH IT
* 🡪 to make this possible: ENVIRONMENTAL sustainability needs to become a NONFUNCTIONAL REQUIREMENT in the SOFTWARE ENGINEERING PROCESS (cite?)
* Need for Transition Engineering
  + Change from unsustainable systems & to more sustainable ones

Why Sustainability Matters

* Definitions
  + Brundtland report – 3 dimensions
  + Goodland -> extended by human dimension -> individual dev. Of humans in their lives
  + Wrt it systems sustainability -> additional technical dimension
* United Nations Earth Summit (1992), UN Rio+20 Conference (2012)
* Focus on environmental sustainability
* Problem too complex for a single mind to solve -> sustainable systems can support
* Sustainable Software Systems can have an impact -> direct, indirect, rebound
* THIS POTENTIAL -> reason why software engineers should care about sustainability!
* Hilty et al. (Sustainability and ICT – An Overview of the Field)
  + Technological efficiency alone -> won’t produce sustainability
  + RATHER in combination with sufficiency
* How to EFFECTIVELY support sustainability through software development?

🡪 LEARN from how safety and security were integrated as NF requirements / quality requirements!

History of Safety and Security

* How did they become NFR’s?
  + Systematically included due to serious accidents / other negative accidents (e.g. incidents were users were hurt / identity theft / financial fraud etc.)
  + -> took a while until they were included in STANDARDS
* Most recent requirements engineering standard - ISO/IEC/IEEE 29148:2011
* P. 4 -> Timeline of Safety / Security considerations
* 🡪 Sustainability: just recently became focus as quality aspect for software

Comparing Sustainability to Safety & Security

* … draw from practices for S & S to derive ways how to treat sustainability as NFR
* Requirement Analysis (as a quality attribute)
  + For goals & objectives -> generic reference model (penzenstadler etc.)  
    + REQUIREMENTS & CONSRAINTS
  + Risk analysis -> Lifecycle Assessment (LCA) approach, holistic!
* Policies & Standards
  + Include content of standards / policies in requirements -> common
  + BUT – these regulatory sources first need to be created -> by extending existing standards

Outlook

* Sustainability should be a non-functional requirement -> but there’s need for
  + Methods for sustainability requirement analysis
  + According extension of policies and standards to include sustainability
  + Quality assurance techniques
  + Sustainability METRICS & assessment techniques
  + 🡪 here SECOMO could support!!!!
* For metrics: base on available metrics
  + ISO 14000 – environmental sustainability (21)
  + ISO 26000 – social sustainability (22)
  + Environmental Sustainability Index (?)
  + 🡪 IEEE 1680 is already taking steps (?) (23)
* + need for STANDARD that give guidance for software development
* Assessment Techniques
  + Proposed: adapt LCA to software engineering & make use of “environmental impact assessment” -> (see Greensoft, they propse the same?!)
  + Either
    - LCA of development process itself, with tools, coding, etc.
    - Or: LCA of operational / usage environment of software – focus would be on business process analysis

Conclusion

* Sustainability in SE is NOT ONLY energy efficiency & green IT
  + BUT also second- and third-order impacts need to be considered!
* 🡪 “To do so, sustainability must be considered as a first-class quality attribute and specified as a nonfunctional requirement of IT systems” (p. 8)
* Focus on environmental sustainability most important – not supported yet (due to author’s opinion)
* And it is so easy to abandon it..
* WHAT is to be done? 🡪 prioritize sustainability in requirement analysis, increase awareness -> SECOMO can support by helping to improve communication & show impacts EARLY
* + sustainability policies and standards need to be created for software engineering
* THEN software can soon contribute to improving sustainability

Exkurs: Sustainability & Software Engineering

* Two ways:
  + Improving sustainability of SE as a process itself
  + Improving software systems’ sustainability in the context or its surroundings -> Sustainability by software engineering
* Software Systems IMPACTS on sustainability -> see Hilty as source
  + First-order impacts -> direct effects of a software system on its environment (cited!!!) (Engery usage, waste production, emissions of infrastructure)
  + Second-order impacts -> indirect impacts / induction effects of software systems -> e.g. CHANGES to user resource consumption / consumer behavior
  + Third-Order impacts -> rebound effects ! -> increased efficiency of systems which lead to users using even more systems 🡪 which then causes even more energy consumption!

## The relevance of ICT for environmental sustainability – Hilty et al. – 2006

* …
* Impacts in 3 orders of magnitude 🡪 see Safety, Security, Sustainability paper (penzenstadler)
* …

### Towards Sustainable Software Engineering (NIER Track) (Green Tracker) – Amsel et al - 2011

* say: SW has indeed an effect on environment, name examples (e.waste, power demands); effects of Software Upgrading (e-waste as old hardware can’t keep up and becomes obsolete, power usage to run the upgrade)
* Definition: “Sustainable software engineering aims to create reliable, long-lasting software that meets the needs of users while reducing environmental impacts”; 🡪 thus meet those aspects where IT – SW has an possibly negative influence on the environment; important for companies that PRODUCE SOFTWARE of course  
  “Sustainable software engineering aims to reduce the power footprint of computers as well as minimize other environmental impacts associated with software systems”
* ICT’s role in harming sustainability: “in 2007 the ICT sector was responsible for 2 % of global carbon emissions” (SMART 2020 report)
* Focus on ENERGY CONSUMPTION of software systems -> show that there are possibilities to change / make things better
* Content of Research: how SE can be sustainable:
  + How do users think about sustainability of software – SURVEY  
    when USING / UPGRADING software, people do not really think about the impact this has on “environmental sustainability”; 🡪 “concept of sustainable software” is not familiar, not considered an important concern, “sustainable software is at best a minor concern”
  + Built their own tool, “GreenTracker, which measures the energy consumption of software in order to raise awareness about the environmental impact of their software using.” 🡪 what happens if people actually see it?
  + “Explore indirect environmental effects of software in order to understand how software affects sustainability beyond its own power consumption”   
    🡪 results:
    - Direct effect: of course power consumption
    - Indrect effects w/in Software: (see Herring: Rebound effect (steam engines)) 🡪 better / improved systems can lead to indeed negative effects, as they are then more popular, more used, and the increased use leads to effects eg. Again overall more power consumtion, more obsolete hardware AND (!! Social !!) e.g. substituting computer/software for humans..
    - Indirect effects in other domains: e.g. on shopping behavior (online shopping) or work life (videoconferencing!!) (🡪 not so much IT but more general)
    - ECONOMIC effect: “economic growth, by enabling corporations to do their work more efficiently” 🡪 3rd pillar 🡪 BUT important to leverage software to use this efficiency for the benefit of sustainable alternatives, otherwise it just accelerates unsustainability as growth needs to be in some ways controlled (SEE this one article!!!! Technocentrism vs. ecocentrism?!)

🡪 hard to say if what the overall effects will be, but need to be aware of it!

* GreenTracker details: Implementation
  + Measure energy consumption by collecting information about the CPU usage
  + For 3 classes (state article) – Autio, Browsers, Word Processing & simulate typical usage
  + Why only CPU? 🡪 no the only aspect for energy consumption!! BUT a key component; has been done this way before(?) 🡪 less CPU power = less power consumption = greener environment (WHAT DO WE BASE THAT ON?)
* Conclusion
  + (what are NFR’s and what is their place in SW lifecycle?)
  + Add sustainability to NON-FUNCTIONAL requirements!!

## !!!!!!!!!! Impacts of software and its engineering on the carbon footprint of ICT – Kern, Dick et al – 2015 / oder ’14?

Abstract

* Energy consumption of ICT is INCREASING
  + Many solutions for Green IT and still progress
  + Not much investigation: software contribution (to energy consumption – negative impacts + positive impacts)
  + CARBON FOOTPRINT – just one way of rating the impact of ICT
* Paper: Calculation Method for the carbon footprint – of software system over its lifecycle
  + 🡪 possibility: compare to SECOMO?
  + + INTEGRATE this calculation in software dev processes (could do the same for secomo?)
  + + discuss impacts & tools

Introduction

* GREEN IT: a lot of work on IT product design & production -> for Hardware + for “green IT service operation” 🡪 past few years
* BUT -> “software as the ultimate cause of hardware requirements and energy consumption shifts only slowly into focus” (p. 1)– not enough studies; not enough approaches that are “capable of being used in the industry” (p. 1) for the software side
* *“Today, there are no practical suggestions on how to monitor and improve software energy efficiency during development.” (p. 2)* 🡪 use as argument support argument?!
* TODO
  + Create AWARENESS and TRANSPARENCY -> SECOMO ☺
* Standard techniques to assess carbon footprint of a product
  + Often “complicated and laborious” (p.1)
  + Often after the product has been released
  + Usually identify parts that are suitable for using renewable energy 🡪 so help to make decisions about changes to renewable energy
* TODO
  + Integrate considerations about sustainability EARLY in the design process

🡪 like SECOMO!!! With estimations!

* + (according to braungart / mcdonough (09) – cradle to cradle / Kramer (12) – User experience in the age of sustainability)
  + 🡪 LATE CHANGES ARE MORE EXPENSIVE -> as argued in secomo!
* HERE:
  + Method that complements (Dick/Naumann (10) – Enhancing SE …) & (Lami et al. Software’s sustainability from a process centric perspective)
  + Focus on
    - Development process itself and its carbon footprint impact
    - Design decisions that impact influence of software product itself
  + 🡪 so that dev team can make decisions based on current carbon footprint!

Method **Carbon Footprint Calculating**

* Calculated based on a number of inputs
  + No. of employees, no. of working days, IT infrastructure, commuting, overall IT infrastructure (enterprise), office space
* Result: in kg CO2 equivalents
* NO CONSIDERATION of “hard to estimate upstream chains” -> 3rd party software
* Strategy: **Continuous Energy Efficiency Measurement** based on Continuous Integration (CI) method + Test Driven Development

Related Work -> reuse for my part???

* ICT, renewable energy, environment
* Green and sustainable software and its energy efficiency
* Carbon Footprint

Integrating aspects of carbon footprints -> the method(?)

Conclusion

* Climate change = one of the biggest challenges -> one cause: CO2 emissions!
* Approaches how to reduce: many; one: Calculate CARBON FOOTPRINT 🡪 to identify optimizationn potential! 🡪 same idea as secomo
* Such a calculation solution is MISSING for software -> **here a solution**
  + Defined an EXEMPLARY ENTERPRISE
  + CALCULATE Carbon Footprint DURING DEVELOPMENT PHASE
  + 🡪Calculate kg Co2 per person month
  + BIGGER SCOPE THAN SECOMO; and other -> process aspect! – but less specific!
* Many impacts during such a phase on carbon footprint, e.g. COMMUTING
* But after all, the more the software is used, the less the actual DEVELOPMENT PHASE IMPACT of the software is relevant (and more the usage phase)
* 🡪 two aspects are important
  + Make the software “production” process greener (with this method – calculate CF of process)
  + Make the product itself greener (e.g. with secomo even one step earlier – improve energy efficiency by adding it as a quality attribute -> see NFR requests etc! -> continuously measure & improve)
* THOSE METHODS HELP IMPROVING SUSTAINABILITY WITHOUT CHANGING SE METHODS IN GENERAL ☺

## Measuring the Sustainability Performance of Software Projects (2010) - Albertao

(Albertao, Xiao, Tian)

* Software is not in itself “environmentally friendly” 🡪 it can have positive & negative aspects, but it is usually not measured during software development.
* Created a method to monitor sustainability, tested on a case, 🡪 to raise “the awareness that software should be developed in a sustainable manner”

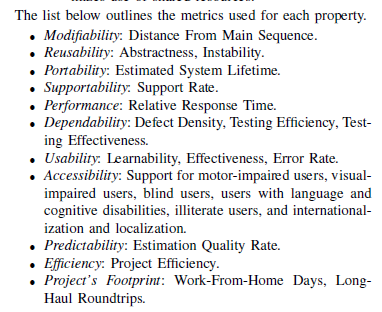
Thoughts on Sustainability & ICT

* Brutland Definition 🡪 + the three areas are interdependent (was also in one of the articles!!)! all dimensions are intertwined
* ICT:
  + “immaterialize activities that otherwise would consume resources” (travel, communication, eg. Mail)
  + Can help to “extract knowledge to optimize resource-intensive processes”
  + IMPACT of ICT:
    - E-waste / hardware that becomes obsolete quite fast – due to NEW SOFTWARE THAT HAS HIGHER REQUIREMENTS

Previous thoughts on Sustainable Software

* Seacord (2003) & Tate (2005): talked already about “sustainability” in connection with software engineering aspects – but “in reality [they are] related to software maintenance, and normally they do not account for the environmental and social aspects”
* Many starting points created: from guidelines for “socially responsible IT practices” by IT companies (though focus on the operation of IT products, not the development), to general strategies of environmental friendly activities applied to IT operations
* 🡪 now: attempt to create a method and metrics to measure & thus understand sustainability performance of software!
* **Definition of sustainability performance: how does a software fulfill certain quality attributes, that have a benefit for the three dimensions of sustainability??**

Method to monitor the sustainability of software projects

* Existing measurements & practices – evaluated regarding benefits for dimensions of sustainability (Economy, social, ecology)
* Method:
  + Assess Metrics when sw is released
  + Analyze the Metrics
  + Identify “Sustainability Improvement Goals” for the next cycle!
  + WHY is secomo enhancing this idea -> estimate metrics BEFORE development, make it possible to improve already in the beginning of the development phase!
* Properties/Metrics analyzed in method:  
  
* 🡪 Metrics are quite “generic”, not so clear how to measure them, no “mathematical models” like SECOMO
* TBC