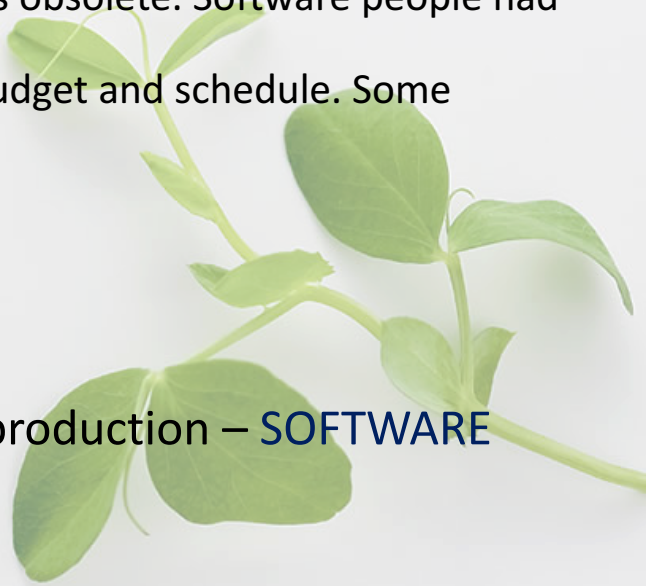
SOFTWARE ENGINEERING

A SUSTAINABLE APPROACH TOWARDS SOFTWARE DEVELOPMENT

SOFTWARE – EVOLUTION

- Focuses on Quality and Productivity.
- Different Eras -
  - The pioneering Era - The most important development was that new computers were coming out almost every year or two, rendering existing ones obsolete. Software people had to rewrite all their programs to run on these new machines.
  - 1965 to 1985: The software crisis - Many projects ran over budget and schedule. Some projects caused property damage.
    - Cost and Budget overruns
    - Property damage
    - Life and Death
  - 1985 to 1989: "*No Silver Bullet*".

This lead to a need for a discipline concerning software production – SOFTWARE ENGINEERING!



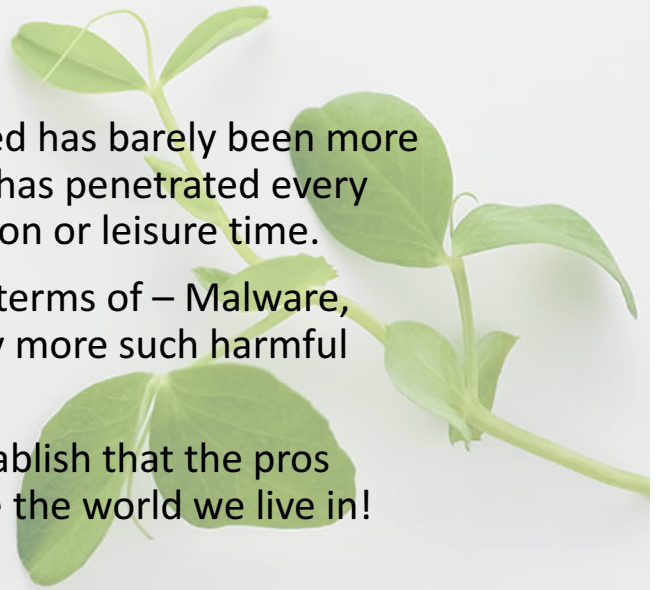
SOFTWARE ENGINEERING – IMPACT

BASIC UNDERSTANDING

Software engineering is the application of engineering to the field of design, development, implementation and maintenance of software in a systematic method.

IMPACT

- The time frame in which computers and software have developed has barely been more than 75 years. Yet, it's impact has been revolutionary! Software has penetrated every field, be it – business, communication, wars, healthcare, education or leisure time.
- While it has simplified lives, it has also increased the dangers in terms of – Malware, viruses, hacking tools, smart weapons, spyware, spam and many more such harmful inventions.
- Pros vs Cons – On weighing the pros and cons, we can easily establish that the pros outweigh the cons, and thus software continues to revolutionize the world we live in!

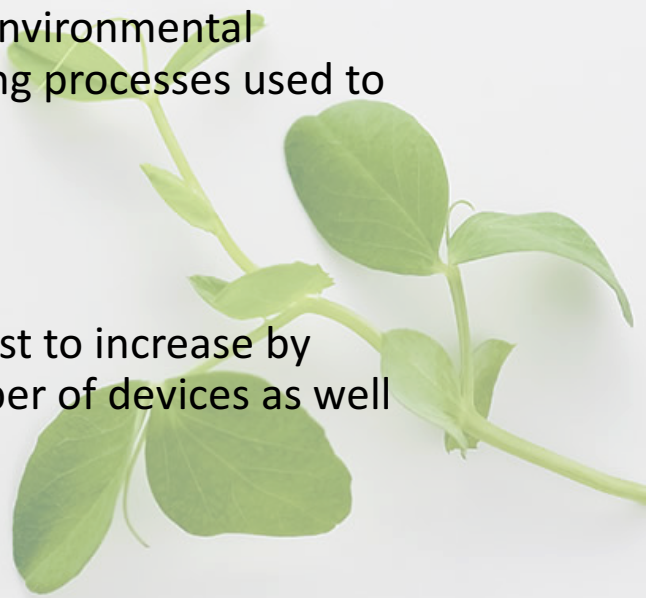


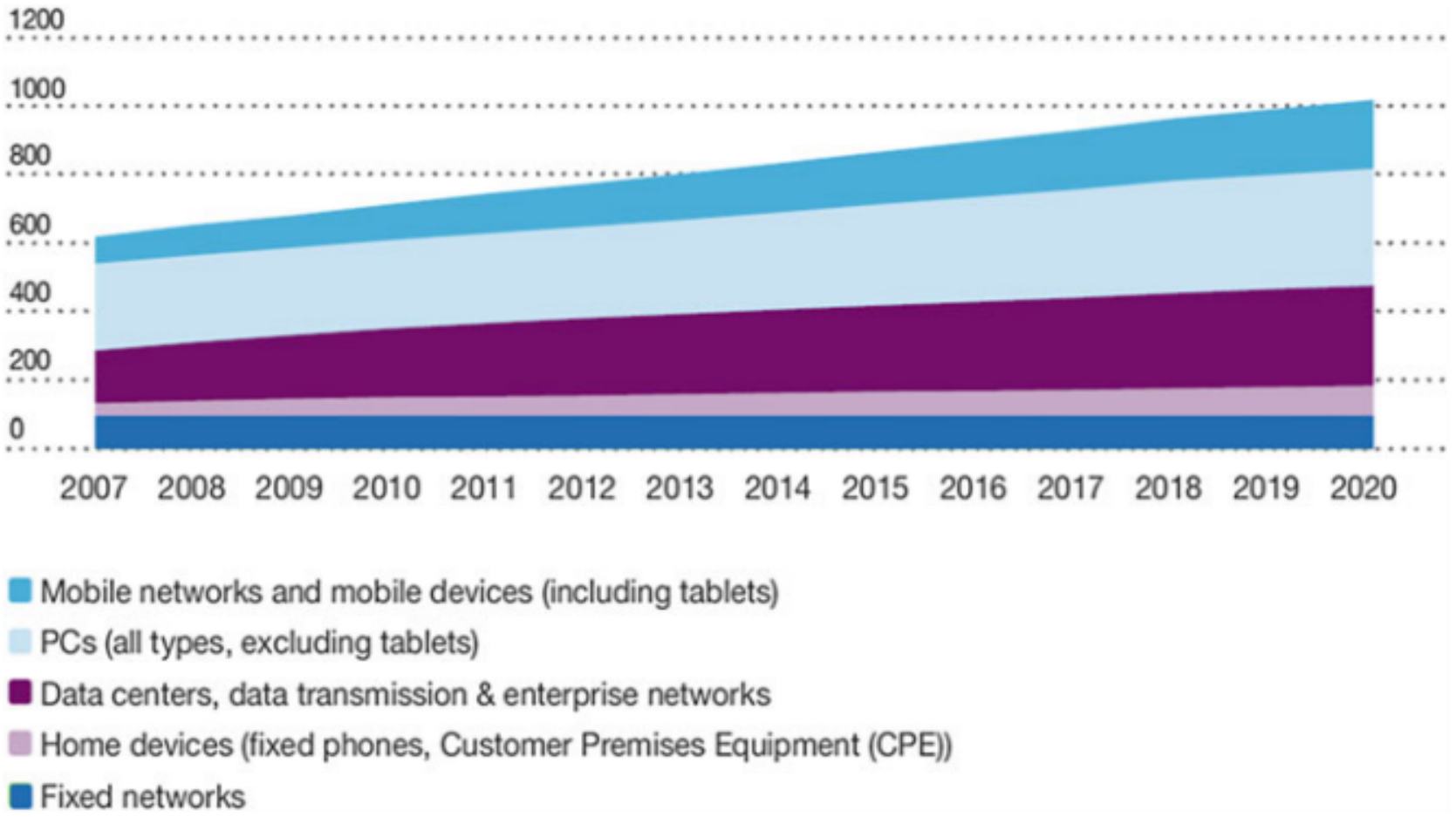
BUT...

NEGATIVE IMPACT ON THE ENVIRONMENT

On the one hand, software helps organizations to tackle environmental issues (using video conferences, dematerialization, more efficient processes, etc.); on the other hand, technology itself is often responsible for major environmental degradation (amounts of energy consumed by the engineering processes used to manufacture products).

- ICT contributes about 2 % of the global CO2 emissions.
- The total electricity consumption of the ICT sector is forecast to increase by almost 60 % from 2007 to 2020 due to the increasing number of devices as well as to network expansion.

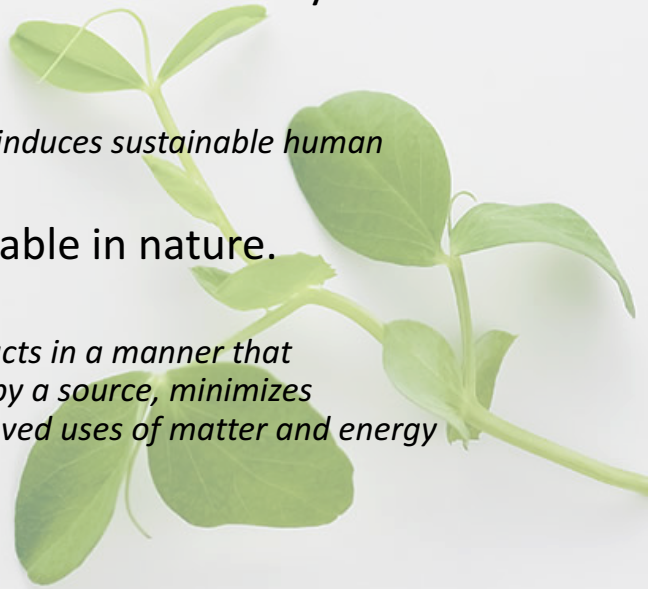



1200
1000
800
600
400
200
0
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
Mobile networks and mobile devices (including tablets)
PCs (all types, excluding tablets)
Data centers, data transmission & enterprise networks
Home devices (fixed phones, Customer Premises Equipment (CPE))
Fixed networks

SOLUTION?

A TWO-WAY APPROACH

- Creating software products that contribute to a sustainable and eco-friendly society.
  - ***SUSTAINABLE SOFTWARE***
    - *Building reliable, self-adapting, low power consuming software that induces sustainable human behavior.*
- Adopting software development processes that are sustainable in nature.
  - ***GREEN SOFTWARE ENGINEERING***
    - *Applying financially and technologically feasible processes and products in a manner that simultaneously decreases the amount of pollution that is generated by a source, minimizes exposures to potential hazards (including reducing toxicity and improved uses of matter and energy throughout the life cycle of the product and processes).*



BASIC PRINCIPLES

1. *Inherent Rather Than Circumstantial* - Designers need to strive to ensure that all materials and energy inputs and outputs are as inherently nonhazardous as possible.

2. *Prevention Instead of Treatment* - It is better to prevent waste than to treat or clean up waste after it is formed.

3. *Design for Separation* - Separation and purification operations should be designed to minimize energy consumption and materials use.

4. *Maximize Efficiency* - Products, processes, and systems should be designed to maximize mass, energy, space, and time efficiency.

5. *Output-Pulled Versus Input-Pushed* - Products, processes, and systems should be "output pulled" rather than "input pushed" through the use of energy and materials.

6. *Conserve Complexity* - Embedded entropy and complexity must be viewed as an investment when making design choices on recycle, reuse, or beneficial disposition.

7. *Durability Rather Than Immortality* - Targeted durability, not immortality, should be a design goal.

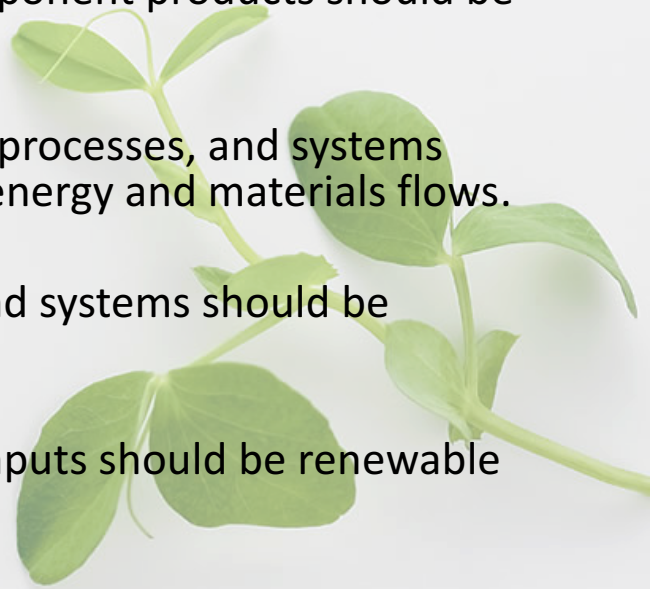
8. *Meet Need, Minimize Excess* - Design for unnecessary capacity or capability (e.g., "one size fits all") solutions should be considered a design flaw.

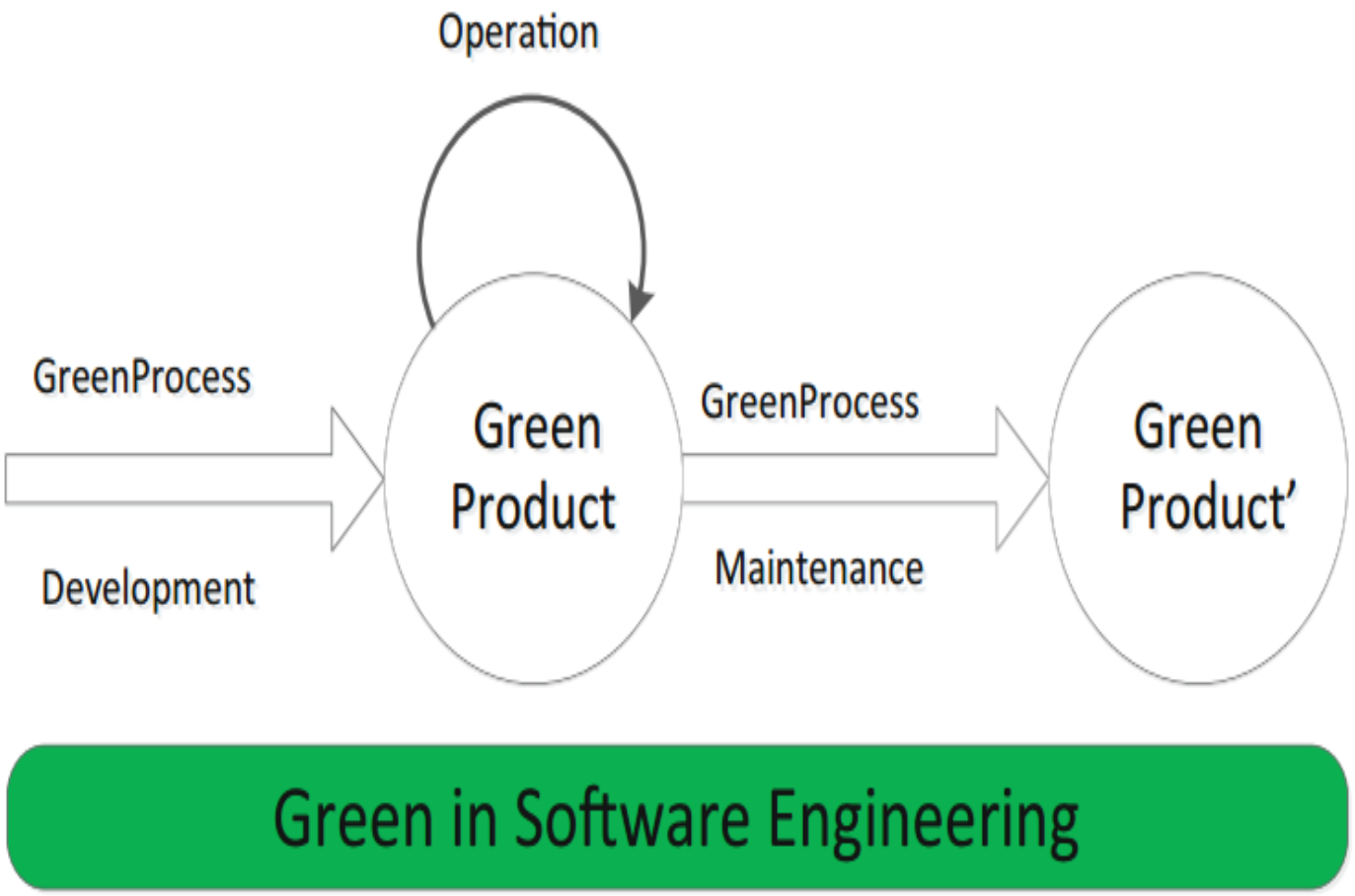
9. *Minimize Material Diversity* - Material diversity in multicomponent products should be minimized to promote disassembly and value retention.

10. *Integrate Material and Energy Flows* - Design of products, processes, and systems must include integration and interconnectivity with available energy and materials flows.

11. *Design for Commercial "Afterlife"* - Products, processes, and systems should be designed for performance in a commercial "afterlife.“

12. *Renewable Rather Than Depleting* - Material and energy inputs should be renewable rather than depleting.



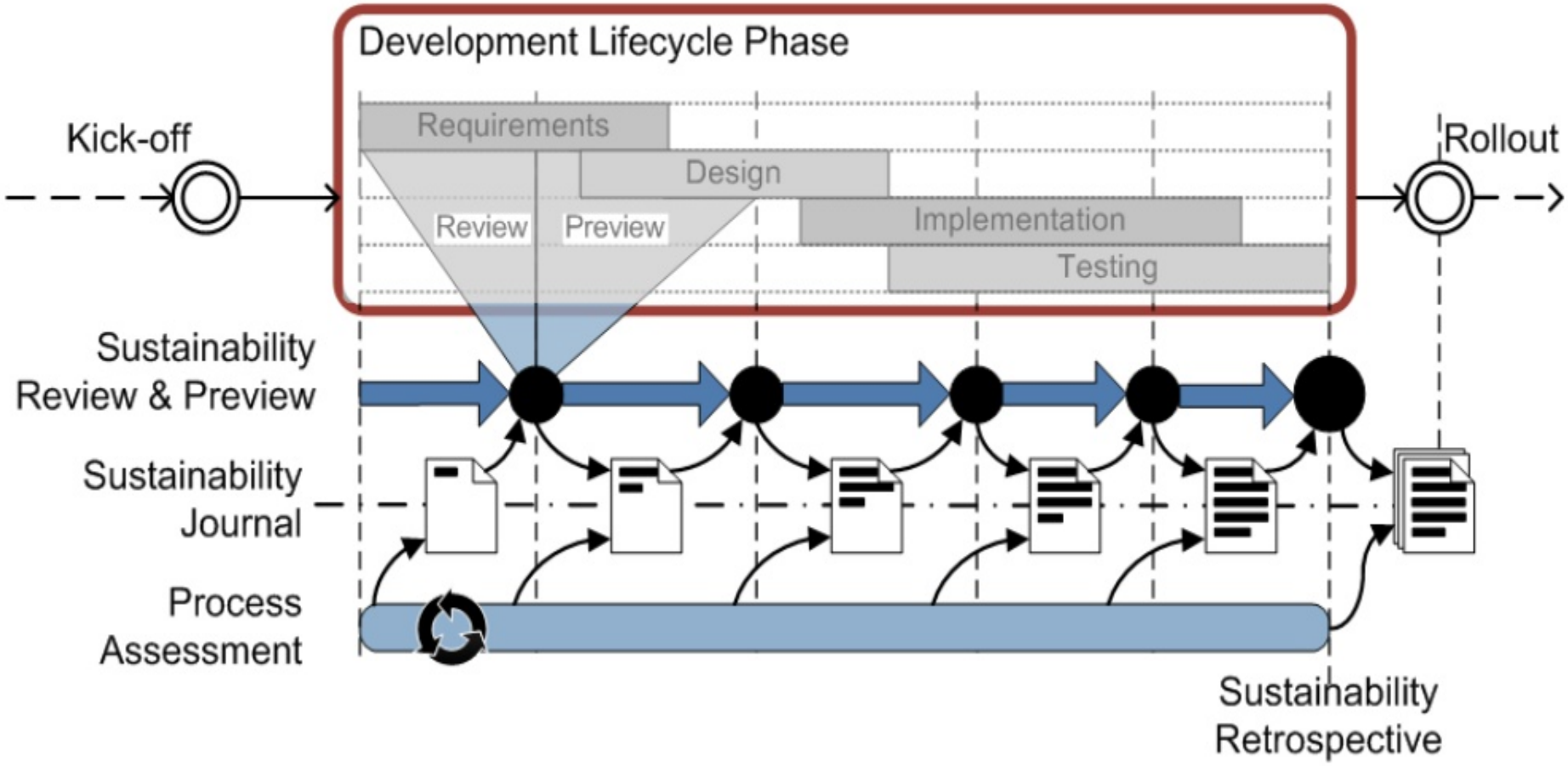

Operation
GreenProcess
Development
Green
Product
GreenProcess
Maintenance
Green
Product'
Green in Software Engineering

MEASURING ‘GREEN’ IN PROCESS

- *The ‘Green’ factor of a software development process can be calculated by monitoring the following aspects throughout the whole cycle, and documenting the observations*.

  - Materials Used
  - Intermediate Outputs
  - Energy consumption
  - Waste reduction
  - Pollution prevention

Overview of a GSE-Process Model



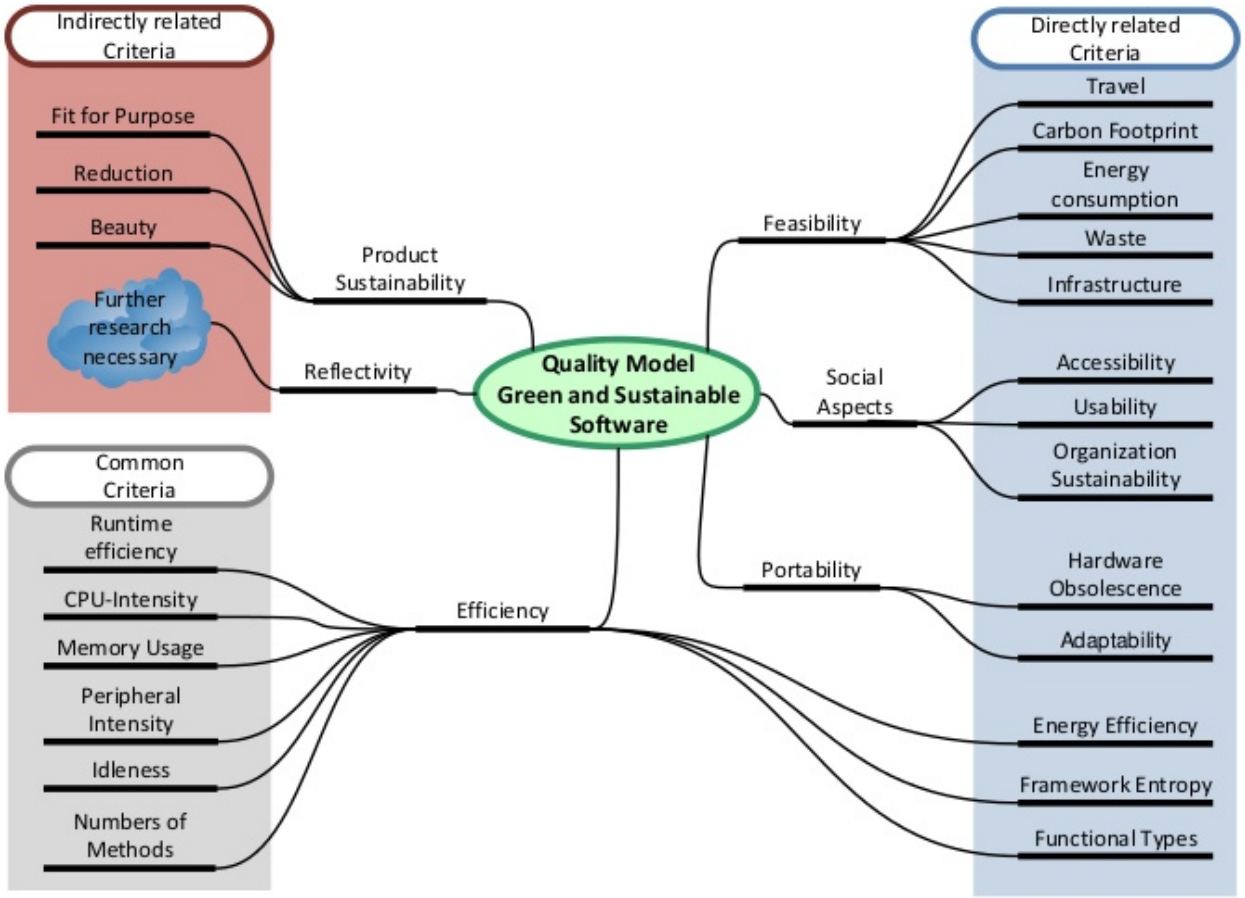
MEASURING ‘GREEN’ IN PRODUCT

- *Sustainability Criteria and Metrics* can be used to evaluate the ‘greenness’ of a software product.

  They can be widely divided into three categories

  - Common quality criteria and metrics
  - Directly related criteria and metrics
  - Indirectly related criteria and metrics

Quality Model of Green & Sustainable Software



CHALLENGES AND DIRECTION

- Challenges faced in Green Software Engineering –
  - Lack of efficient green hardware and resources
  - Lack of skills & manpower required for green engineering
  - Lack of energy consumption measurement techniques
- Direction of Green Software Engineering –
  - Developing a standard green model
  - Empowering programmers to adopt to ‘green’ methods

REFERENCES

- ‘Green Software Engineering’ book by Carol Calero and Mario Piattini
- Green Software – Slideshare by GreenSoft Company
- Wikipedia –
  - History of Software Engineering
  - Software Engineering
  - Green Engineering
  - Sustainable Engineering
- Green Software Engineering – The curse of Methodology – Research Paper

THANK YOU!

Presented by -
Poojita Suri
ASU2013010200049