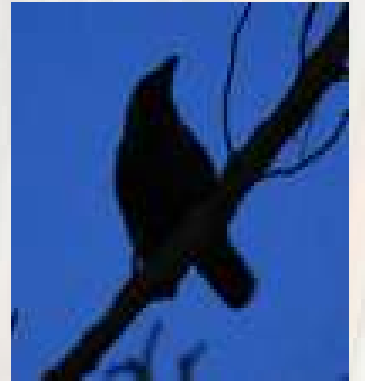


Applying the
Principles of Free Software to
Software Procurement, Development,
Deployment and Maintenance

abhas abhinav



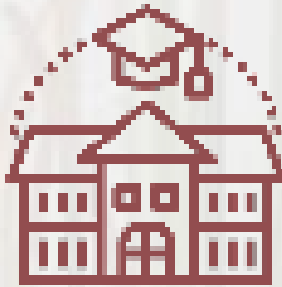
abhas.io



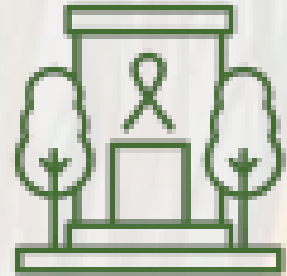
To create the context of what
I want to talk about, let me
start with 3 stories:



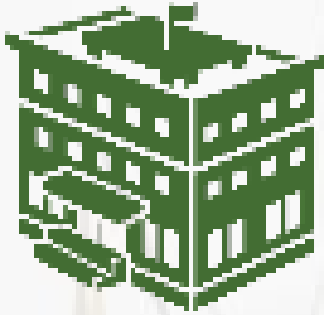
Public Sector
Tech Company



Univeristy



Social Enterprise

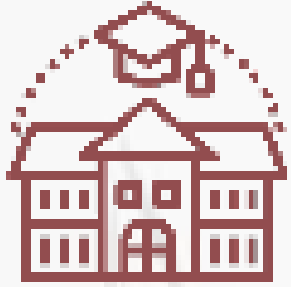


Public Sector
Tech Company

Target: use free software to build a service for the public

Goals: better product, low cost of development + deployment, efficiency, security

Method: outsource development + implementation to a private sector software firm

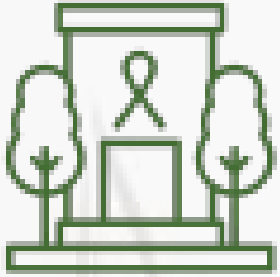


Univeristy

Target: use free software for system to evaluate exam answers papers

Goals: productivity, efficiency, tech-assisted process, scale

Method: outsource development + implementation to a private sector software firm



Social Enterprise

Target: use free software to build an digital learning platform

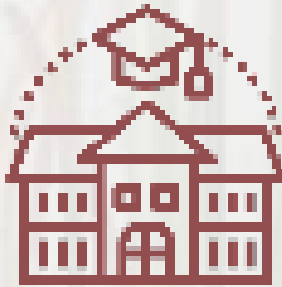
Goals: time-to-market, security, scalability, features, best technology

Method: outsource development + implementation to a private sector software firm

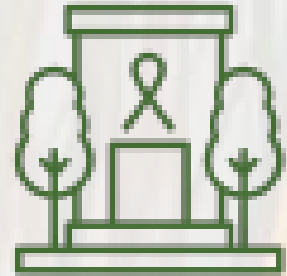
Results did not serve goals.
Could not take advantage of Free Software.
Loss of time, opportunity & funds.



Public Sector
Tech Company



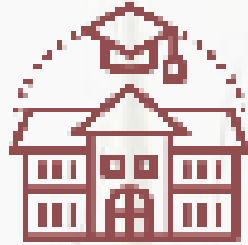
Univeristy



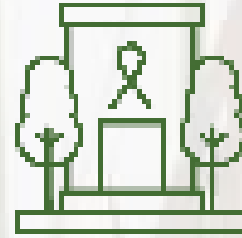
Social Enterprise



Public Sector Tech Company



Univeristy



Social Enterprise

No source code! Lack of knowledge about the implementation. Low awareness about free software principles, licensing, ethics and practices.

Actions did not deliver intentions.
Free software got blamed!



It is up to us to make sure that
our choices and actions are
consistent with our goals.
Free software enables this.



How could we avoid this?

Awareness of free software principles. Vigilant and mindful application of these principles. Corresponding involvement in the process. Focus on self-reliance.

Desirable Long-term Goals?

Ability to scale?
Privacy we deserve?
Ability maintain it
ourselves? Opportunity
to extend it? Integration
with other systems?

Transparency?
Building competency?
Opportunity to become
self-reliant?
Sustainability?

The principles around which Free Software is built, shared and used gives us the opportunity to learn, to cultivate ownership, to have privacy and to build competency and hence, the ability to maintain, extend, integrate and sustain systems ourselves – all in a very transparent, auditable and reproducible manner.



Choosing Free Software



Source code. Version Control.
Packaging. Testing.

Reproducibility. Automation.
Documentation. Communication.

Opportunity & Freedom

Reproducibility

Can you build it yourself?

If you built it yourself, would it have the same results? Is it possible to “verify” someone else’s build? Can you compare your build with others’? Does repeating the same steps provide the same set of deterministic outcomes?

Verifiable Auditability & Determinism

Continuous Integration

Demonstrates consistency, reproducibility and automation. Eliminates *ad-hoc* delays between detecting problems → fixing problems → testing & packaging solution → deployment → user testing

Continuously Demonstrable Deliverables

Build Competency

Technical awareness + involvement.

Document and insist on a value system: processes and principles.

Transparency & auditable trust.

Ability, Ownership & Understanding

Build Competency

No longer somebody else's problem.
Shared, responsible & true
ownership. Transparency.
Sustainable choices.

Independence, Freedom and Self-reliance.

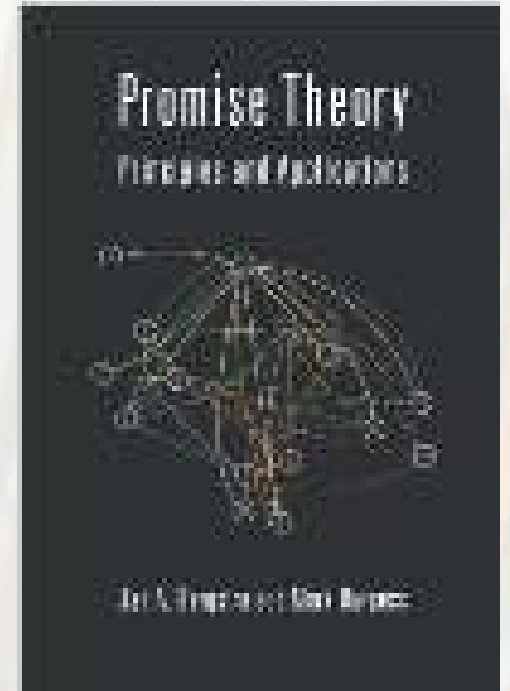
Management

Multiple vendors. “Vendor pipelines”.
Measurable consistency between
choices, methods and goals.
Technically verifiable “promises”.

Processes, Communication & Trust

“Promise Theory”

We are “agents” and we make “promises” to others in the “system”. The promises should be “verifiable”. A “promise” does not guarantee an “outcome”. It needs to be clearly documented otherwise it's not a promise. “Conditions” should be a part of your “promise”. The “language” of promise must be shared. It needs to be “mutually agreed” (no obligation) otherwise it's not a promise. You might depend on other “agents” to **your** promises. Other agents make promises to you. But you can not make promises on “behalf of others”. To keep a promise, you should have a choice. Every promise binding is the basis for relationship.



For U & Me

Open Journey

Software Freedom Can Lead to Self-Reliance, says **DeepRoot Linux Founder**

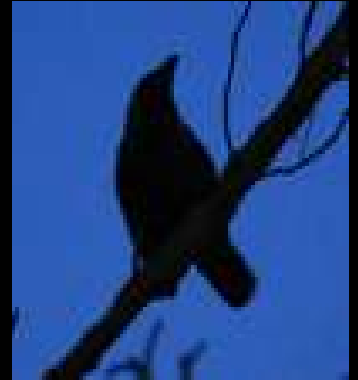
<https://opensourceforu.com/2019/02/software-freedom-can-lead-to-self-reliance-says-deeproot-linux-founder/>

Freedom and Self-Reliance are,
after all, what we are all looking for.

THANK YOU!

“What I cannot create; I do not understand. Know how to solve every problem that has been solved.”

Richard Feynman



abhas.io

