

Timeplan

Author: **Tor Nilsson Öhrn**

Program: **Medieteknik**

Title: **Git as a product life cycle information management tool.**

Introduction to the problem

Git is, according to Susan Potter, in its structure not limited to exclusively manage source code history. A manufactured physical product could possibly, in a way not very different to source code, change after it has left the store. Just like source code, products can be considered to be "cloned" by customers when they are bought. Information about the product then diverges over time if it is damaged or reviewed by the customer who bought it. Later on the customer's version of the product history can converge with the manufacturer's updated version and create a merged version containing more information about the product. Could Git be used to manage this kind of dynamic product information? In which ways can Git be changed to better manage information about physical products? How could this kind of product repositories be used as a persistent ad for our belongings?

Preliminary way to solve the problem

A prototype will be created to demonstrate how git could solve a use case scenario where a customer buys a product on-line, adds information about that specific instance of the product and then finally sells it on a second-hand site later on. In the thesis it will also be discussed though how Git could be used by manufacturers, resellers and recyclers of products and where it perhaps falls short.

Prototype

At first it should be evaluated if it would be possible to just encapsulate Git as it is in a client program that can be used by end-users for product life cycle information management (PLIM). If this approach is not satisfying and if more time is available one could create a git fork that better follows the requirement of a PLIM system. One of the main issues will probably be how to represent a repository for a specific instance of a product, i.e. the repositories that end users will maintain. The focus of the thesis will be on the so called middle of life (MOL) information of a product, i.e when the product is already at the end-user. The prototype will not cover use cases in the phases where shipping, manufacturing and recycling are involved, i.e beginning of life (BOL) and end of life (EOL).

Evaluation

Kary Främling [Främling et al., 2009] suggests that the success of an information architecture for physical products and the implementation of it can be evaluated by the following criteria:

- Degree of fulfillment of the requirements set up by the demonstrator owners (i.e companies interviewed for Främling's report but could in this thesis be changed to a end-users requirements for using the repository as a second-hand ad).
- Scalability.
- Flexibility of the system when organizational, hardware or other changes occurs.
- To what extent it satisfies the requirements of an Internet of Things.
- To what extent it satisfies the requirements of PLIM.

These criteria are explained further in the paper and I should be able to use Främling's criteria as a reference for the thesis when evaluating Git as a PLIM system.

List of References

- C Corcelle, K Framling, L Rabe et Al
Assessment of item-specific information management approaches in the area of heavy load vehicles
[link](#)
- S Potter
the Architecture of Open Source Software Applications - vol 2
[link](#)
- S Chacon
Pro Git
[link](#)
- C Ranasinghe, M Harrison, K Främling et Al
Enabling through life product-instance management: Solutions and challenges
[link](#)

Timeplan

The programming of the prototype will be divided into 3 sprints with time inbetween each of them for reflections and report writing. Below is a preliminary time schedule organized by week numbers:

- **6** Thesis concept decided and found articles about the subject.
- **7** Read articles about ISIM and Git
- **8-12** Off duty
- **13** Set up and document requirements for the prototype based on articles from Främling et al.
- **14** Choose platform, frameworks etc. for the prototype
- **15-18** Programming of an initial prototype. (SPRINT 1)
- **19** Half time check
- **20** Off duty
- **21** Evaluation of the initial prototype. Was an unmodified version of Git sufficient? Document conclusions in the report.
- **22** Either keep on prototyping by refining the initial prototype or fork Git's source for more fundamental modifications.
- **23-26** Programming of a second version of the prototype. (SPRINT 2)
- **27** Evaluation of progress so far and documentation of conclusions in the report.

- **28-31** Programming of a third and final version of the prototype. (SPRINT 3)
- **31-35?** Preparations before presentation as well as the actual presentation

Half time check goals

At the half time check there should be some kind of software ready where even a not very tech-savvy user can:

1. Retrieve product information from an (reseller's) remote repository to their local item specific repository (ISR) which now should represent an unique instance of the product.
2. Take a photo of the product and add it to the ISR. *OPTIONAL*
3. Pull and merge updated information of the product from the remote repository upstreams into the ISR.
4. Push the ISR to a Github like server for second hand selling.

The design choices of the prototype should also by now be documented and motivated in the actual thesis paper.