

Reflection on Practice

1. Taylorism

Frederick Winslow Taylor,
author of,
*The Principles of Scientific
Management* (1911)



Defined the idea that work could be *measured* and that worth and utility could be *assessed* by scientific/mathematical principles.

This spawned *time and motion studies* that still are the major driving force in much of modern workplace design and practice.

So, a worker must:

- have n cubic metres of space
- put in 40 hours of work per week
- be in the office
- keep a formal record of all work done

An employer must:

- enforce uniformity of practice
- keep formal records of this practice
- remove individuality

2. The Paradigm

A *paradigm* is a way of thinking, a **mindset**. Taylorists are thralled to the Taylor paradigm. Having a paradigm enforces you to see the world in a particular way: **through that paradigm**.

Paradigms (or mindsets) are extremely powerful:

- Blair v Cook / Bush v Chirac
- Christianity v Atheism
- Capitalism v Socialism
- "*I hate ...*"

Breaking out of a paradigm into a new one can be traumatic and is often - indeed, should be - life-changing. **E.g. St Paul on the Damascus road.**

3. Reflection

Donald A. Schön

author of,

The Reflective Practitioner (1983)



A new paradigm ... a new way of thinking

Schön noticed that,

- native peoples learned skills in an applied way
- there was little formal teaching involved
- education had become *Taylorist* wrt teachers and learners
- learners and teachers were practicers
- learning came from practice

Q: How did this learning work?

A: Schön proposed:

- we do not learn complex skills from books or words
- best practice is best *done* (i.e. do and learn)
- we build up a **practicum** of skills that cannot be learnt in a classroom
- *ticking boxes* - classic Taylorism - is no measure of the quality of work
- you cannot assess or increase work quality by observation - you need to be also **involved**

This process of learning by doing, increasing your practicum of skills, he called **reflection**.

3. Reflection and the Software Engineer

Programming is a task in the classic line of master craftspeople: musicianship, painting, stonemasonry, teaching, ...

There is

- a body of knowledge - programming principles, etc
- an individually crafted end-product
- no right answer definable in advance
- no single right answer
- lots of wrong answers
- a shared body of knowledge and skills
- that is had to define on paper
- etc.

A **reflective software engineer** will

- keep a realistic record of what (s)he is doing
- will observe inter-personal relationships *wrt* work quality
- will **reflect** upon personal practice with the aim of learning to improve
- will **reflect** upon team practice with the aim of becoming a better team-player or manager

4. Issues about being Reflective

To reflect you must be **involved** - you cannot improve practice without **practicing**

e.g. medical lecturers, art teachers

Being reflective is part of the response group against such as Taylorism, rote learning and deep versus shallow learning.

A paradigm change often produces illogical hatred and alienation from the old paradigm:

- Object orientation v structured approaches
- Sony v Microsoft v Nintendo

Reflection gets easier and better understood the more it is practiced.

-> see also papers in CA506 area