An Exploration of the Phenomenology of Software Development

Charles Tolman
Who would usually rather be flying!

Why Explore Phenomenology?

- And just what has it got to do with Software?
- Our technological world has flowered thanks to the Cartesian, dualistic world-view. This has allowed us to develop modern science, giving us the industrial revolution. Though a necessary process - there are negative consequences.
- Software development has heralded a major change in the form of tools we use. This is bringing the flaws in the Cartesian world view into sharp relief.
- Phenomenology can heal the split world-view and the negative consequences. Software development can benefit from this different perspective.
- But we need to be aware of the different approaches of the East and West cultures in making this unification.

TALK STRUCTURE

- Background : Novice – Journeyman - Old Grump.
- Key Observations : Boundary Crossing.
- Philosophy : Descartes – Goethe.
- Software : Knowledge – Imagination - Patterns.
- The Future: How to heal 'The Split'.

Background: Novice – The Early Years

- Started out with Electronics at 17 back in 1974!
  Originally a shy young adolescent 'nerd'.
  Found comfort within the 'inner world' of thought.
- Southampton University Electronic Engineering Degree in 1979.
  Moving swiftly from hardware to software developer in 1980.
- Moved into the field of Media. TV/Film Editing systems.
  Worked with cool tech and was enticed by the faerie glamour of the toys!
  Implemented a multi-tasking scheduler in Assembler.

Background: Journeyman - The Dangerous Years!

- Playing with more complex, generic, structures. Many of which don't get used.
  Focusing on the tools rather than the problem.
  Creating unnecessarily complex solutions.
  Letting the **Idea** overshadow the **Problem Context**.
- Optimism & arrogance about what can be done, soon followed by…
  SHEER PANIC as the system gets away from you.
- Lack of realisation that debugging requires more complex thought than coding.
- Highlighting the fact that the programmer needs to become more self-aware in order to progress through this phase.

Background: The Grumpy Old Programmer

- Very aware of the limitations of one's thinking.
  - One is too frequently WRONG! Again and Again and Again and Again…
  - Particularly easy to see when debugging.
- As maintenance becomes a priority there is a drive to simplicity/clarity/minimalism.
  So the code can look like novice code but will use complexity when required.
- This Appropriate Minimalism shows the need to find the TRUE ESSENCE of a problem.
  What I call 'Japanese Garden Software'.
- Yet there needs to be a balanced judgement between Perfection and Pragmatism.

Key Observations: The Importance of Energy

- It takes a significant amount of mental energy to do this job.
- We are making many decisions minute to minute.
- The difference in the rate of progress when there is more energy.
  As a technical lead a significant concern is the level of ENERGY in a team.
- This equates to Alexander's comments about Life in Architecture.

Key Observations: The Foundation in Play

- Many early programmers were amateur electronics/radio enthusiasts.
  Then the amateurs turned professional.
- At the beginning of the PC 'revolution' many programs were games.
- Transition from programming as PLAY to programming as ECONOMIC.
  But we forget this foundation in PLAY at our peril:
  - It is a fundamentally human activity that fosters resourcefulness.
  - It helps us to understand the world.
  - And of course is FUN!
- Of course there is always a need to balance:
  - Play & Economics
  - Quality & Utility
- But the scales have tipped too far away from Play.

Key Observations: Boundary Crossing

- Tools/Technology are Amplifiers.
  Industrial Revolution: Devices to amplify physical capabilities – and mistakes!
  Information Technology: Devices to amplify thought – and mistakes!
- This is a transition from **EXTERNAL** to **INTERNAL** problem solving.
  It is a **significant** shift and has many side-effects – we need to be **more** awake, not less.

THIS HAS LARGELY GONE UNNOTICED!

- Robert Glass' Fact 13:
  "There is a disconnect between management and their programmers."
- This means that we are not 'just' programming and will need a larger view of the discipline in order for there to be constructive progress.

Key Observations: The Inner World

- Programming involves the creation of a completely human-made world.
  With little feedback and few checks against external reality.
- Alexander's comments about Life in Architecture relate to physical space.
  Though are dealing with a MENTAL space, his world-view is highly relevant.
- The process of software development can be seen as follows:
  - We engage in THINKING.
  - We create MENTAL MODELS or THOUGHT CONSTRUCTS.
  - We transcode these into SOFTWARE.
- A good programmer needs to be SELF-AWARE about this Process and their own Learning.
  The computer can be seen as a 'Thought-Mirror'.
- Initiation in Thinking: formerly mystical/religious experience. But now religion is not adequate.
- Taking on ideas from Eastern traditions as-is is not the best. East and West are different!

Philosophical Interlude: Some Dudes From History

- Bacon 1561 – 1625
- Descartes 1596 – 1650
- Newton 1642 – 1727

- Goethe 1749 – 1832

Philosophical Interlude: Descartes : Dualism

- Descartes : 1596-1650
  - Cleared the decks to be sure of what we can really know.
    Cutting down to the minimum by getting away from the senses.
  - Split between Mind/Body.
  - But trying to fit in with the church's religious ideas.
    e.g. 4th Law of Impact!
  - res cogitans : Thinking – Mind/Soul : Active.
  - res extensa : Extension – Body/Nature : Passive.
  - Subject vs Object

Philosophical Interlude: Goethe : Natural Scientist

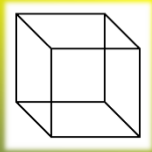
- Goethe : 1749-1832
  - Around during the Industrial Revolution.
  - Natural Scientist as well as a Poet.
  - Concerned by the Baconian approach of the separation from, and the mastery over, the natural world.
  - Goethe was working with natural phenomena.

Philosophical Interlude: Goethe : Philosopher

- Goethe : 1749-1832
  - Focus on the phenomenon - not on abstract ideas.
  - Warned of the danger of over-hypothesizing…
  - The need for a **Delicate Empiricism** during observation.
  - The importance of **Exact Sensorial Imagination** - not ungrounded fantasy!
  - Rudolf Steiner (1861-1925) realised that the same methods could be used for thinking. i.e treating **Thought as a Phenomenon** in the same way.

Philosophical Interlude: Phenomenology

- The process of 'Coming into Being', i.e. Appearance (verb), of a Thing.
  A very difficult concept for us to understand.
  But it is this idea that deals with, and heals, the destructive consequences of the Cartesian subject-object split.
- We need to understand that the Appearance and the Thing are ONE item.
- The primary enduring insight about brain function [McGilchrist]:
  RIGHT: Immediacy of Lived Experience. Appearance
  LEFT: Re-Presents what is Lived – in order to Know. Thing
- To know something new requires us to split concept and percept (Descartes) and then re-unify them (Steiner).
- We **cannot** continue to keep focusing on Results as primary.
  [Agile does have the beginnings of this focus on process - but primarily as an externalised idea.]



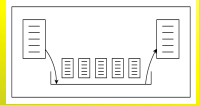
Software: Knowledge or Preconceptions?

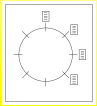
- The 'Tracer Bullets' experience of knowledge generation highlighted the dangers inherent in over-hypothesizing.
  It is too easy to become automatic and habituated in one's thinking.
  [Workshop at OT2001 facilitated by Paul Simmons/Tom Ayerst]
- Just how aware are we about:
  - The limits of our own knowledge?
  - The process by which we expand our knowledge?
- How often do we jump to conclusions rather than carefully collect data or information – whether it be debugging or problem analysis?

- **A DELICATELY EMPIRICAL method is a MUST HAVE when dealing with faults or user requirements.**

Software: Imagination

- Generally we believe we are in the geometric domain when considering software structures…
- But is it geometric?





- REALLY?
- Or are we thinking in terms of FLOW such that the images drop away…
- And how fit are the structures we are imagining i.e. 'building'.

- **A DISCIPLINED IMAGINATION of structures and behaviour is a MUST HAVE for good design.**

Software: Christopher Alexander & Patterns

- I agree with Jim Coplien that we need to "revisit the Alexandrian roots".
  Although Agile is based upon Alexander's core values we have not properly embraced his idea of patterns. [See http://www.youtube.com/watch?v=VV4KXKa5ZdQ]
  Alexander seemed bemused at OOPSLA 96 and then, at a later talk he gave in Stanford, sceptical about prevailing perspectives in the software industry.
- There is a difference between **True Liking** and **Apparent Liking**.
  Alexander uses the concept of the **Mirror of the Self** test to tell the difference.
  True Liking is an art and a skill that needs developing.
- If these ideas are developed we can get away from the subjectivity of 'Beauty is in the eye of the beholder'.
  But we need to put ourselves into the picture.

- **We need to regain the perspective of PERSONAL DEVELOPMENT implicit in Alexander's world-view.**

The Cartesian Split: Or Goethean Phenomenology?

This underlying dynamic of 'The Split' occurs in many places:

- Descartes: Subject & Object.
- Brain Function: Left & Right Hemisphere.
- Career: Playful Child & Clever Adult.
- Problem-Solving: Abstract Idea & Grounded Perception.
- Judgement: Perfection & Pragmatism.
- Knowledge: Analysis & Synthesis.

This Split is healed by truly understanding the UNITY of APPEARANCE & RESULT that Phenomenology talks about.

A VISION FOR THE FUTURE : A CHOICE

We can either:

A. Allow humans to become harnessed to the technology, so they become just a more effective Software Development Tool within a mechanistic world-view.

OR

B. Consciously develop ourselves both personally and technically, so we can better shape the technology towards truly human needs, raising the awareness of both developers and users.

Todo list if you want to take it further:

- READ:
  'Taking Appearance Seriously' by Henri Bortoft.
  Probably the best introduction to Goethe's method and Phenomenology and a very good historical overview of the pertinent philosophy.
- STUDY:
  'The Nature of Order' by Christopher Alexander.
  A lot to read but a master work giving insight into Alexander's world-view.
- READ:
  'The Master and His Emissary' by Iain McGilchrist.
  Another excellent master work dealing with brain function.
- DISCUSS:
  Get in touch!

Thank you

Blog

charlestolman.com

Email

ct@acm.org