



Käre läsare,

I LPD-gruppen sysselsätter vi oss med tankar om lärande, kunskap och konsten att skriva effektiva texter och egentligen hade jag tänkt skriva något om detta, men tänkandet känns inte färdigt så det får istället bli en text som jag skrev för några år sedan om en viktig princip i dynamisk produktutveckling.

## The 80/20 rule

“To do it right the first time, as other theories prescribe, is a very expensive way to work. In DPD, we instead try to do almost right, that is, achieve an 80% solution in our first try. With the same ambition in the second try, one quickly achieves an almost 100% solution.

In order to be able to use the 80% rule, developers must have several problems to work with concurrently. They then iterate between these problems achieving 80% solutions in each loop. If we on the other hand have only one problem at a time to work with, we will automatically try to achieve 100% solutions, which is both expensive and a waste of time”.

Professor Stig Ottosson

The 80/20 rule is also called the Pareto principle. Here is how it works. Initially the work result grows almost linearly with time, figure 1. Then, as we get closer to work finished, efficiency diminishes because our knowledge gets less valid the further we go. There is a knee on the curve beyond which progress grows ever more slowly in an asymptotic movement towards 100% finished. Work should be halted at the knee, at approximately 80% of work finished, for the following reasons.

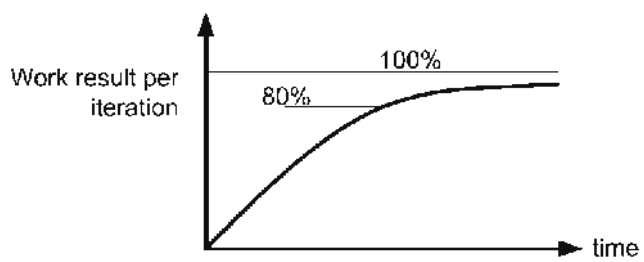


Figure 1. The 80/20 rule, just do 80% right the first time around, then 80% of the remainder in the next iteration, etc

- ❑ The value of knowledge often decays with time. The knowledge we start with is less valid the longer we keep on without replenishing with new knowledge (verify results, coalesce with the rest of the team, etc).
- ❑ Therefore the probability, or risk, that we will have to go back and redo earlier work increases progressively the further we go.
- ❑ Furthermore, the cost of design changes increase the longer we continue without replenishing our knowledge.
- ❑ Risk cost is defined as probability of change multiplied by cost of change. Therefore, risk cost increases very steeply beyond the knee of the curve, figure 2.

It is good practice to stop at roughly 80% finished and then shift to other tasks and in this way verify what one has produced and replenish ones knowledge. A break is also good for ones creativity. During the break, we get new impressions while at the same time the preconscious mind works with the problem.

The 80/20-rule actually is just as much a 90/10 or 27/73-rule. The exact numbers are not important, but the thinking, the principles behind are,

- Tempo, initiative and money is lost if people spend their time waiting. Therefore, you should not stop working. If you for some reason cannot continue with what you have at hand, then shift to something else until you are able to go back to and continue the first work task.

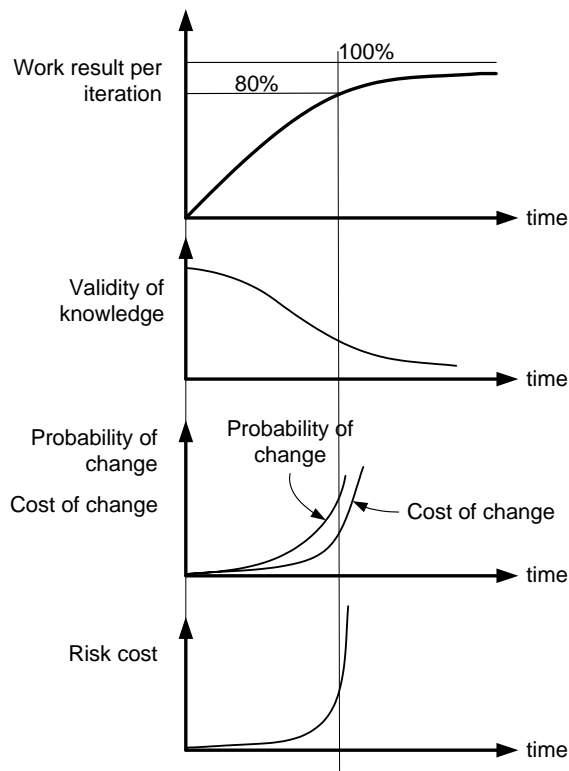


Figure 2. The Pareto principle in DPD

- If the reason for your stop was a lack of ideas, the problem perhaps required creativity, then the break and your rest from precisely that problem will lead to an incubation in your mind. Your preconscious mind works with the problem and the next time you come back to it you will find that you have actually made progress. This is one reason why you do not lose time by helping a teammate with her problem for a while.

So your creative capability benefits from this switching between different problems or work-units. Switch not only between design problems but also between design, prototype building, testing, observing users, planning, etc.

How many problems can you handle simultaneously, or rather consecutively in a repeated iterative way? This depend on experience, simultaneity capacity, if you are tired, how you

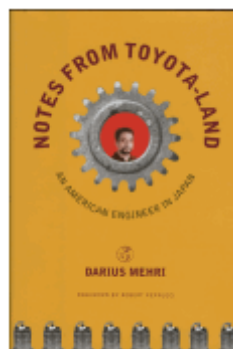
handle stress, etc, but surely somewhere between two and ten.

In this fashion, using the 80/20 rule, you can proceed cycle by cycle until the design is finished.

## Veckans bok

**N**otes from Toyota-land är boken du måste läsa om du är det minsta intresserad av Toyota Production System (TPS) eller Lean Product Development. Darius Mehri är en ingenjör från USA som arbetade tre år på ett företag i Toyota-gruppen i slutet av 90-talet.

Mehri beskriver ingående hur det är att arbeta i Japan, hur fackföreningarna fun-



Darius Mehri,  
**Notes from Toyota-land: an American engineer in Japan**, (hardcover), ILR Press, 2005, ISBN 978-0-8014-4289-6 (231 pages, 233\*161\*20 mm)

gerar, eller inte fungerar, hur hierarkierna styr, hur alla tjuvhåller på information av rädsla för att bli bestulna på sina idéer, och hur falsk den västerländska bilden är av det påstådda beslutsfattandet via konsensus.

Vidare beskriver Mehri hur beroende de nyanställda är av sina chefers välvilja för att lära sig utföra sina arbetsuppgifter eftersom Japanska universitet försummar att ge sina ingenjörstudenter nödvändiga praktiska kunskaper.

Sammanfattningsvis, detta är en mycket läsvärd bok som du absolut bör läsa. Rekommenderas.

Boken får ★★★★★.

Lars Holmdahl • +46 708-77 53 16 • lars@larsholmdahl.com