# DPD rules of thumb

## Use pull, instead of push

***It is better to use pull than push, partly because with pull you implicitly obtain the right direction no matter where you are.***

This has several implications as regards human systems. If we see push as the whip and pull as the carrot, we understand that it is much better to set up a system to rely on pull rather than push as its governing principle.

From basic courses in mechanics, we may remember that two forces, $F_1$ and $F_2$, are identical if their magnitudes are equal and they act along the same line, figure 15-1.

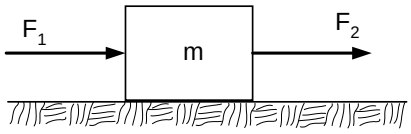


Figure 1. Pull and push

In reality however, this is not the case once the mass m has started to move. If we rely on push ($F_1$) we must constantly adjust the direction of the force in order to have m move in the desired direction. This is not the case if we rely on pull ($F_2$) since in this case the direction is implicit. Compare for instance the act of balancing vertically a rod in your open hand by pushing against its lower end, to letting it hang freely from its upper end.

The ramification for PD is that it is much better to guide with a vision than to govern with detailed checklists and timetables. A vision as a goal will communicate a direction at all times. Should the need arise for a change of directions of a PD project, then it is much easier to move the vision slightly than to rewrite specifications and timelines.

## In every moment do the most important

***To achieve maximum efficiency, the shortest time-to-market (TTM), lowest cost, etc., you should occasionally stop and ask yourself, "is this the most important thing I can do right now."***

What is the most important? Here everyone's judgment comes into play. A simple answer to the question might be that which leads to the goal, which best realizes the vision. This once again shows the importance of a common situational awareness, that all team members share the same vision and works towards the same goal.

## Ask yourself, "how do I/we now that?"

***In product development, we should base our decisions on as good, objective, and truthful picture of reality as possible. This is what is normally meant by the term: "fact-based decision making". Instead, perceptions and decisions are surprisingly often based on various unsubstantiated narratives that freely circulate within the organization, or based on personal prejudices and pet ideas. To counter this, you should ask "how do we know that," in cases where the decision data and its quality is not readily apparent from the context.***

Unverified knowledge can have many roots. It can stem from traditions ("how do we usually do?" as a basis for action) and policies that once may have been meaningful, but since lost relevance, are still followed out of habit. In this way, an organization is controlled by its history, which can be dangerous when exterior conditions change.

## Focus on the process, not the result

***We know from manufacturing (lean production) that you cannot get high quality by just measuring the quality and try to control it. Instead, one must measure and control the factors that give rise to high quality.***

***It is the same in product development. It is too late to control based on the outcome because it often comes after many days, sometimes months of work. Therefore, one must find other factors to base control on. Experience shows that a process approach in the form of attention to the individual's everyday activities produces good results.***

## The user—not the customer

***Put users/usage in focus because without user focus the development is easily ineffective. Ideally, a demanding user should be part of the picture during the development process. Product developers need to understand the situation of users thoroughly before, during, and after the development process. This is done preferably with own user studies and conversations with users.***

***Start from BUS: During product development, it is important to always ensure that usability (U = user/use), the company's business (B = business) and impact on society (S = society) are included in the assessment. Customer requirements, as many theories value the most, are only a subset of B that primarily sales people have to consider.***

One often reads that one must have a customer focus**Fel! Bokmärket är inte definierat.** in product development. We hear that you should create value for the customer. When asked, what customer? Is it the wholesaler? You get the answer: the end customer! However, the end customer, for example of a car is the scrapping company and steel mill that makes reinforcing bars. Surely, it is not for them that we develop the car?

At a certain car manufacturer, they say that they develop their cars for the owner. They develop not only for the first owner, but also for the second and third owner, because resale value is important, to the first owner.

A significant problem with starting with a "customer requirement" is that the customer rarely knows what he wants, and especially not in the case of radically new innovative products, "wish-driven" product development, or so-called "technology push".

Research has shown that the product will be better if you develop it for the user. Sometimes, customers and users are the same person, but often not. Take the example of a taxicab. The owner of a taxi company is a customer and in a sense, he can be said to use the car. The cab driver is quite clearly a user, but so are also the taxi company's customers who travel in the taxi. It gets even more complicated when the taxi gets a special transport operation paid for by the municipality.

It is wise to think through who the users are and in what manner they will use the product. For it has been shown that if you have a user focus, you will pay more attention to user values.

To understand the user better, you should become user yourself.

## Switch between tasks—do not rely on a specific preplanned sequence

***Unlike other theories on product development, it is permissible and even recommendable to jump between different tasks. Development controlled by a checklist means wasted time. However, a checklist may be consulted afterwards to ensure that no important points have been missed.***

Your creative ability will increase if you switch between different tasks. Tempo, initiatives and money will be lost if people spend their time waiting.

If for some reason you cannot continue with what you are doing, do not remain idle, but switch over to the second most important until you can resume what you were originally doing. New research has shown that the more experienced and skilled the designers are, the more they iterate between activities.

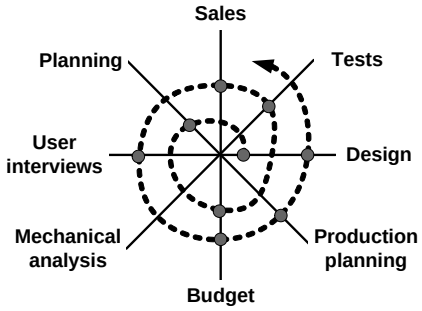


Figure 2. Do not spend your time waiting but shift between tasks

Contrary to other PD theories, it is recommended in DPD to jump between different tasks because that will increase your creative capability. While you are resting from a problem that very problem is processed by your preconscious mind yielding an effective "incubation".

## Make many small, and few large decisions

***To maintain maximum speed, one must constantly make new decisions exactly when needed, and avoid saving them for later, when they become big decisions, or making decisions prematurely before you have requisite knowledge. The effect of a few big decisions will be waiting for many, and difficulties in correcting erroneous decisions even when reality shows that this should be done. It is also generally not dramatic to change small decisions, while the opposite is true for large decisions made at a higher level. Other developmental theories advocate few and big decisions, i.e. fine grained long-term planning, which reduces flexibility in a harmful way.***

The dynamic approach is based on communicating a clear goal, a vision, which gives the work direction. With a free flow of information, shared knowledge, and situational awareness, governance by communicating a vision, gives necessary conditions for relying on a large degree of self-organization.

It assumes a rough long-term plan and a constantly updated, more detailed short-term plan. Further, by aid of the latest gained knowledge we continually make many small decisions that do not need to be exactly right, just take us in the right direction (80/20 rule).

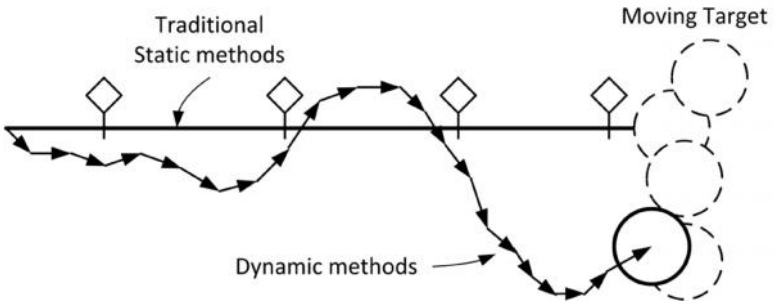


Figure 3. Often in PD, one is aiming at a moving target

The dynamic method could be called the gate-less method, or every minute a gate if you will. The opposite is stage-and-gate methods such as Integrated Product Development (IPD), Stage-Gate (SG), and as Concurrent Engineering (CE) is often organized.

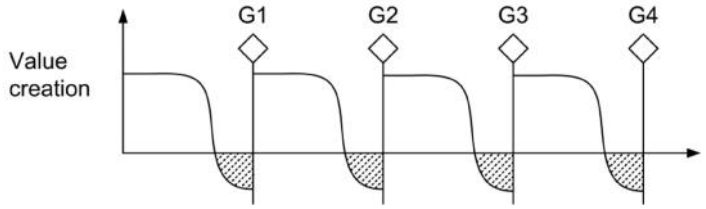


Figure 4. Before each gate efficiency drops as useful work goes into preparation (up to 25% of the total) for the gate, which is the big decision point

In stage-and-gate methods, gates (toll gates) are decision points for major decisions. In very large projects these are often used politically by the project manager in two ways:

- the project leader extorts resources from the line organization (if I do not get resources, then I will not pass the next gate), and
- at a gate passage, the project leader is absolved of all sins committed in the previous phase.

The disadvantages of gates, and their accumulation of decision making, is that they give an illusion of control and that they slow down work, and raise costs.

## Be like flowing water

***In dynamic product development methods, your ambition is to constantly move forward. Like flowing water, pass by small obstacles and leave them for later, concentrate instead on solving the primary problems, like flowing water that is stopped up and accumulated at major obstacles until eventually, it finds a weak spot and breaks through.***

This is a principle in the form of a metaphor based on water's flexibility or "formlessness" as it smoothly adapts to the circumstances and on its momentum when it rushes forward.

*Like flowing water, we should be fast, flexible, and opportunistic. This flexible use of arising opportunities is only possible if the team is fully informed and aware of the overall goal, objectives, and what is to be delivered by the project*. It follows that the project manager must encourage and allow initiatives from his project members. Mistakes are always allowed if they are the result of meaningful risk-taking.

Focus on the main problems and pass by lesser problems, like water flowing around smaller obstacles, without solving them just yet, or setting them aside and solving them separately from the main work, perhaps by aid of a special task force.

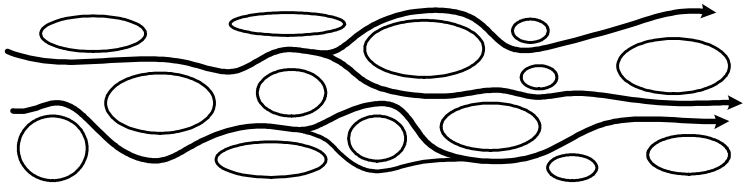

Figure 5. Be like flowing water

The key is the flowing water's flexibility and momentum. If the obstacle is massive, water accumulates and eventually finds a weak spot and breaks through. In the same way, larger, perhaps critical, problems are attacked and resolutely solved with the combined force of team members and other project resources. Do not waste time and resources on secondary problems until you have made sure that critical problems have been solved.

## Identify and focus on the main problem

***Identify the main problem, the big hindrance that the project stands or falls with, and attack it. When the main problem is solved, it is often easier to solve the lesser problems. If you cannot solve the main problem, then you should not waste resources on the smaller problems, but stop or redirect the pro-***

***ject. This is contrary to other theories that prescribe the identification of numerous "requirements" that are compiled into long lists.***

This rule is a subset of the preceding rule "be as flowing water".

## Identify primary and secondary demands

***When the primary demands have been met, it is important to find secondary demands. They are for example service friendly solutions, easy assembly/disassembly, recycling aspects, etc.***

It is important to study primary and secondary users in order to see how users actually use the product, and to get impulses for further development. First you should become a user yourself and try-out the product. Thereafter you should observe users and talk to users. It is often said that by talking to, or observing, at least eight users you can catch the majority of problems with the product.

## Make sure that the primary and secondary demands support each other

***Many primary products need secondary products for their proper function. When the primary product is sufficiently developed, the development of the secondary products must start. Examples of secondary products are transport packaging and service tools. Other examples are supports, foundations, protective hoods, and battery adaptors and other power sources.***

## Maximize functional- , sensuous-, and image values

***In product development, designing the basic function usually consumes most of the time and resources. However, that the product functions as expected is taken for granted by users and customers. This is not the quality that excites customers and persuades them to purchase our product. It is sensuous (touch, smell, sound, texture, experience) and image values that set a product apart from the competition. Therefore, maximize the functional values, the sensuous values, and the image values.***

In fact, if users use their senses when using our product, then the developers must use the same senses when developing the product in order to safeguard the right experience of the product.

## Switch between totality, module, and detail

***In product development, we must start with the product as a whole and then drill down to detail, but also jump back and forth between the two scales during the remainder of the work. Tests are first carried out on detail level, thereafter on module level and finally on the complete product.***

***It is important that the developers manufacture prototypes and carry out tests by themselves in order to gain maximum learning experience, minimize knowledge losses due to hand-offs, and keep a high tempo.***

## Simultaneously develop in parallel the product concept and goals

***Other theories advocate a progression of fact-finding, followed in sequence by analysis, solution generation, action planning, action and finally tests. This is a very time-consuming and expensive way of working, because it does not allow for learning and adaptation to shifting circumstance.***

***In dynamic theory these activities are therefore all performed simultaneously, i.e. new data are analyzed immediately and provide a basis for action that immediately is tested and corrected. Thus, a very high tempo can be sustained both in development and in learning.***

***Similarly, we develop the product concept continuously by always using the latest gained knowledge. The product concept is not a static document, but must be developed throughout the product life cycle. The basis of the product concept is either a task or a challenge (that is a problem or an opportunity).***

It seems to be natural to most of us to first collect data (market research, product spec, etc) and then analyze the data and only thereafter generate solutions and/or develop concepts. This is what several theories advocate. Moreover, it is the method taught at colleges and universities.

However, it is an expensive, time- and resource wasting way of working. That is why in dynamic product development these tasks are performed in parallel, figure 15-6.

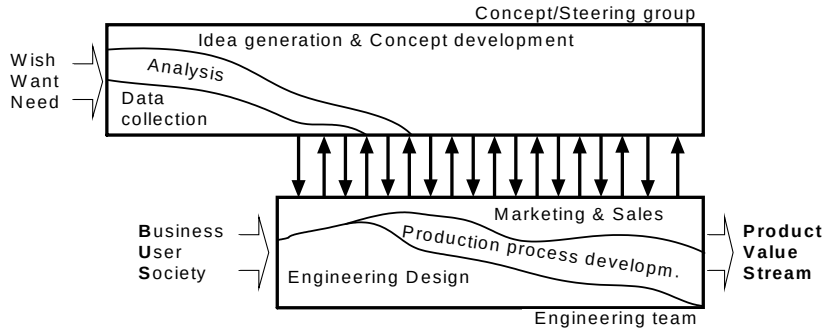


Figure 6. Dynamic product development is an intense learning process that make use of the latest gained knowledge

New data is immediately analyzed, produces actions, that are tested, analyzed, and gives rise to new corrective actions, etc. (The difference between a skilled interviewer and a survey you might say.) In this way, one can achieve very rapid learning and development.

Through these rapid iterations, or concurrency in parallel activities, where questions yield answers that generate new questions and answers, etc., one can reach a very high effectiveness and efficiency.

As you during work, gain more and more knowledge of the product and its users, you will be able to create an increasingly better fit between product and user by allowing the product specification to develop in parallel with concept development.

This adaptation results in better products more in tune with user requirements at product launch. The concept is "owned" by the Concept Group, which is also the Steering Group of the project. The Concept Group members are present in the team for the whole length of the project, which avoids waste due to hand-offs, and creates an important core team that carries the team's history and identity.

## The 80/20 rule

***To do it right the first time, as other theories prescribe, is a very expensive way of working. With the dynamic philosophy, 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, the third, etc, 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***

The 80/20 rule is also called the Pareto principle. Initially the work result grows almost linearly with time, figure 15-7. Then, as we get closer to work finished, efficiency diminishes because our knowledge gets less valid the further we go.

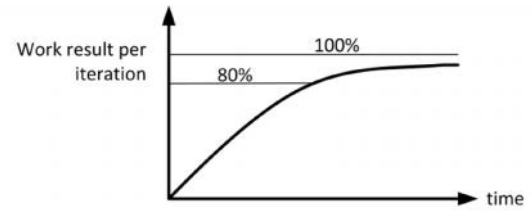


Figure 7. The 80/20 rule, just do 80% right the first time around, then 80% of the remainder in the next iteration, etc

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, figure 15-8.

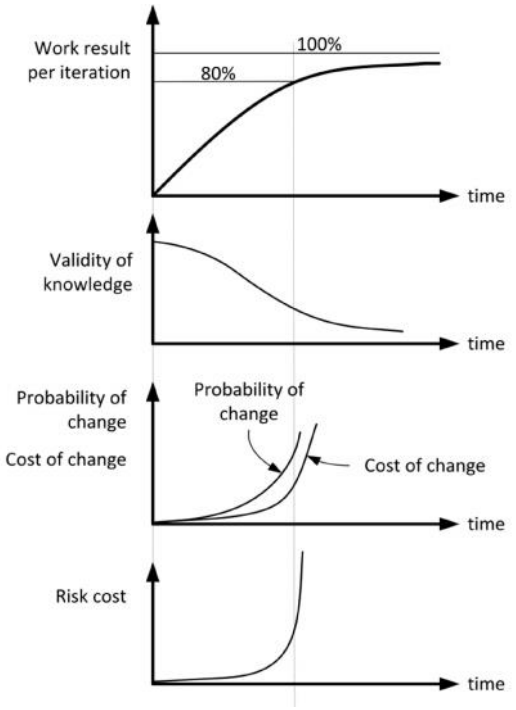


Figure 8. The Pareto principle or 80/20 rule

The reasons for this are:

1. 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).
2. Therefore the probability, or risk, that we will have to go back and redo earlier work increases progressively the further we go.
3. Furthermore, the cost of design changes increase the longer we continue without replenishing our knowledge.
4. 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 15-8.

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 one's knowledge. A break is also good for one's creativity. During the break, we get new impressions while at the same time the preconscious mind works with the problem.

## Taking notes—keeping a diary

***In product development, it is vital that each product developer takes daily notes in their bound notebooks on how the project progresses, and their thoughts and understandings. The diaries are valuable when problem occurs. The diaries also have a legal value, for example, in patent disputes. Diary notes stored in a computer has little legal value!***

Every team member should take notes that are kept in tidy order in a binder regarding:

- Communication and decisions:
  - minutes from meetings
  - telephone conversations (write up: who, date, time, what was decided)
  - fax and email
- Their own work:
  - ideas
  - calculations
  - test results
  - observations

There are several reasons why each individual in a development team, besides taking daily notes, should also keep a diary.

By keeping a diary of decisions, why they were made, etc, it is easier to go back to earlier solutions in case that should prove necessary

Patents are sometimes challenged. Then it may be important to be able to prove when the invention was made

Liability reasons demand the taking of notes. The notes will show that every precaution was taken to hinder unfortunate outcomes from using the product. In addition, the notes can show that the technological level at the time of design was not higher than manifested in the product. This fact can be important since it has been found that people quickly forget the fast technological progress and thus demand the same performance of yesterday's product as they do of today's, especially in a court of law.

For a diary to have legal value it must be written using pen and ink or a ballpoint pen. Pages should be numbered and be regularly signed by another person and, further, pages should be bound together so that pages cannot easily be added or removed.

## Design and verify concurrently

***In order to achieve maximum learning, effective use of resources, and optimize ones designs while minimizing risk, one should design and verify concurrently.***

Not many years ago engineering design of the part/system/total architecture, was followed by the building of prototypes. The prototypes were then tested and test results were analyzed. Often there was no time for redesign in case of failure during test, so the design engineer designed the parts sturdy enough to pass the test. This, of course, was a waste of raw material, money, and a missed opportunity for learning.

This is case A in figure 9.

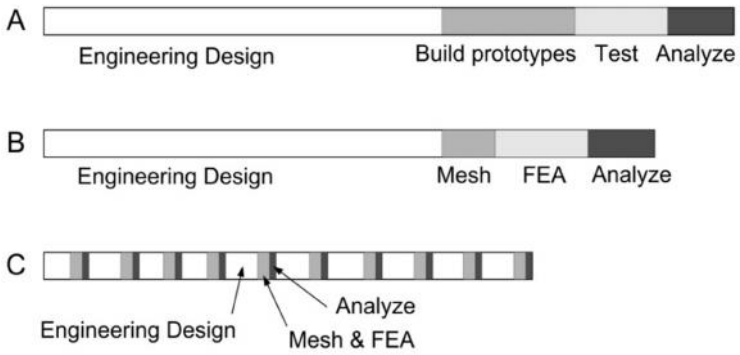


Figure 9. Different development strategies in foremost the automotive industry, A: old-fashioned, B: contemporary, and C: modern (Holmdahl 2003)

When CAD was introduced, CAD drawings and later CAD models were used for making test specimens. After some time it was realized that, the models could be meshed and used for FEA-purposes, making the physical testing obsolete[1]. This is case B in figure 9.

Unfortunately, case B has the same drawback as case A; the waste of raw material and money. The reason for this is simple: in both cases, physical tests and FEA are used at the end of the process to verify the design, not allowing any iterative design loops for reason of time shortage. Findings from FEA were not fed back into the design process.

The preferred method in dynamic product development is case C, which consists of short iterative design-FEA-analyze steps. By using modern software such as DesignSpace® from ANSYS that works in the background of the CAD program and automatically creates the mesh, the engineer can test and modify the design many times during a single day.

By starting out with a coarse FEA-model and making it finer and more precise as the design itself is developed, it is possible in most cases to have the design verified the very instant that it is changed or a feature added. With this method, there is no need for a special validation activity after design is finished, because the design is already optimized as regards strength, noise and vibration, fluid dynamics, etc.

The method of case C allows for quick iterations. This is fortunate because the second time around, the designer will perform faster and better than the first time. For each iteration, the designer gets to know the product and its characteristics better.

There is an additional meaning to the idea of starting with a coarse concept and then refine it in subsequent steps, sometimes iteratively, and that is that you develop the concept continuously from start of project until finished product. This runs contrary to the established paradigm in engineering design, but is never the less a more efficient strategy for developing products that fit the market situation at product launch. The mindset should be characterized by a preparedness for continues concept development.

**Remark.** The above discussion used FEM as verification method. Other simulation tools and methods may be used of course, such as simple tests, comparison with previous experiences in the form of trade-off curves and check-sheets, etc.

## Create a vision and rely on coarse long term planning with a rolling detailed short-term plan

***If not all involved in a development project have a shared vision of what should be the outcome of the project, activities will be unfocused and aimless. Especially the project leader must carry the vision.***

***Due to product development complexity, it is useless to draw up detailed long-term plans. These should instead be relatively coarse, conceptual, while near-term plans can be detailed.***

[1] This has not completely happened, and will never happen as long as legislation in many countries demands physical tests, such as crash tests of automobiles.

It is important that the project leader makes available to the team members all documentation that describes the product concept and project progress in such a way that they can easily grasp its content.

A free flow of information that is openly discussed leads to shared knowledge. Further, with a common goal and common situational awareness, we can, if there is trust between all concerned, achieve self-organization. Self-organization is the most effective and efficient action mode of any organization.

Project control should consist of team members making detailed plans for the coming one to two weeks (other theories advocate detailed long-range plans, which only delays start-up and stifles flexibility).

## Weekly reporting

***Unless the team use visual planning (VP), or as a supplement to VP (especially during concept development), each product developer should make short weekly reports (a few lines suffice) and distribute to everyone in the team in whatever format is most convenient.***

***The reports shall show: 1/ What I will do next week, 2/ what I did last week, and 3/ how much time, money and other resources have I consumed.***

The project leader should with these reports as a start, and in combination with his observations, write and keep a project diary, so that the project can be studied afterwards and lessons can be learned to benefit future projects. If the project leader finds it hard to write, he may be better off having a project assistant that handles such matters.

Weekly reporting has several advantages:

- it provides the project leader (PL) with information about the state of the project in addition to what the PL learns from "management by walking around" (MBWA);
- it provides the team with an opportunity for reflection and analysis: have I really done the most important things?
- it provides a weekly documentation of the project that is easy to distribute to organizations outside the project, the project sponsor, steering committee, etc.

An additional advantage is that it is always better to let knowledgeable people lead themselves (self-organization) than to give detailed orders about what people should do.

## Colocation

***Team members should be colocated in open spacious office landscapes, in which the product is placed in the middle. This makes contact paths short and operations optimally concrete. If team members are sitting separated, each in their own room or cubicle, efficiency and creativity will be reduced.***

The importance of colocation cannot be overstressed. Projects failing to yield expected results can be ascribed to this fact. In addition, the physical layout of the building is important. We shape our buildings, and afterwards our buildings shape us.

Teamwork depends on continuous communication: "*In colocated teams, team members frequently report that some of the best discussions occur spontaneously, based on frequent interactions with colocated workers*". Through all channels: 1/ hearing the tone of voice, words used, 2/ seeing the body language, clothes used, skin hue, 3/ scent: humans pick up and are affected by pheromones, 4/ tactile information, etc.

Humans have a bandwidth of approximately 10 Mbit/s when meeting face to face. All of this, except less than about 20 bit/s, is subconscious communication.

Colocation of the team has many benefits.

- The team stays focused.
- No stealing of team members' attention from other groups,
- short communication routes,
- easy to have impromptu meetings,

- facilitates overhearing, yielding efficient spreading of information:
  - if for instance, the project leader talks over the phone with the sponsor, then the team, by hearing the conversation is informed;
  - this also makes possible spontaneous problem solving which can happen when one team member hears others talk of a problem to which he happens to have a solution.

When colocating the team, the product, or a model, mock up, etc, of the product to be developed should be placed in the center of the group. This has many advantages:

- works as a reminder of the reason for the team's existence and helps focus attention to the product,
- is a good visualization aid for talks and discussions between team members themselves and between team members and visitors to the team, and
- shows the status of the project if the most resent version is displayed.

Colocation of team members mean that they all sit in the same open room. There should be no obstacles between them that block visual communication.

Colocation also means that the team leader, project manager, or chief engineer, must never "hide" in their room isolated from the team. Their place is in the center of the team.

## Make use of the walls

***Use the walls to fill the room with project information, pictures of users in their environments, prototypes, sketches, etc.***

The visual planning board should not be hidden away in a corridor, but hang on the wall so it becomes visible to all team members. Then it will serve as a constant reminder.

It is advantageous to hang up essential information such as:

- description of the product's users,
- images of users, use of the product, and
- pictures showing the styling and the environment in which the product will be used.

Drawings such as assembly drawings, sections, and fitting surfaces are plotted and hung on the wall along with pictures, A3s, and sticky-notes showing regions of conflict, unresolved problems, alternative concepts, etc. So that whenever a team member leans back to raise their eyes they fall on something related to the project thereby filling their brain with visual information that feeds the creative process in the subconscious.

This method is extremely powerful, but it seems almost impossible to convince people to use it. Not until they have actually tried, can one convince them.

## Use projects for product development

***A project is the most effective and efficient form of organization for product development from the first concept sketch to production start (at least for wish- and want-driven PD and in the short term even for need-driven PD).***

***The more team members, the more difficult the communication within the team (with n team members, there are n(n-1)/2 internal connections). Therefore, a team should have a maximum of 6-8 members. If the project team needs to be larger, the project should be divided into sub-projects, each with their sub-project leader that is a contributing design engineer, not an administrator.***

In most cases, development projects are formed within a line organization, thus forming a matrix organization. Depending on how this is done, we get a lightweight or heavyweight matrix, or shades in between.

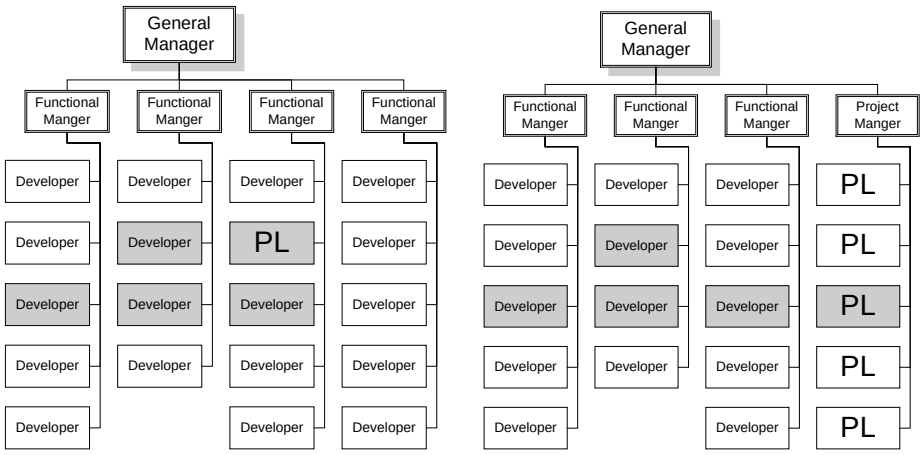


Figure 10. A light weight PL with his team (left) and a heavy weight PL with his team (right)

One problem with the weak matrix, which gives a weak project leader (PL), is the difficulty for PL to compete with subordinates' line managers for subordinates' attention. Since it is line managers that hire, fire, and determine the salary and other benefits. To some extent, this is true also for heavyweight project leaders, but by virtue of belonging to a section of project leaders only, they become stronger versus the line manager.

The most efficient form of organization for new product development (NPD) is a *tiger team*, which is composed of members from different parts of the company (and possibly also from suppliers and customers) and co-located in a separate building. By handpicking members, one can build a very potent team.

## Designing with extremes

***To solve a well-defined problem (a limited number of parameters and precisely specified parameter values) you concentrate on solving exactly that problem. However, in order to increase your knowledge about how a product should be designed, it is appropriate to select and study different extremes. If you have solved the problem of how the product can serve two extreme cases, you most certainly have solutions also for the cases in-between.***

Product development is about finding good enough solutions to problems. The development engineer very actively makes many decisions. To guide the decision-making we have traditionally relied on product specifications.

Typically, a specification is a noun and a number. For example, mass < 2 kg. This method is preferred in many cases because it is easy to handle and it is easy to verify compliance with the design specs.

Such specifications may seem convenient to use but are seldom sufficient. First, you rarely know what specific metrics provide the best compromise, or if the specs are free from contradictions (one can easily over determine the product), and secondly it works poorly for the development of prestige products. To develop such products, we should base our design decisions in an attractive narrative and a guiding vision.

In order to find the best solution, it pays to consider the worst cases, the extremes, and visualize extreme usage of the product. When designing manual and electric window regulators for an automobile, we visualized a timber jack in northern Sweden trying to wind down a frozen window, the plastic handle must not break. Concurrently we visualized a small female driving her children to school; the winding torque must be limited to allow easy maneuvering of the window.

By thinking of the extremes and solving for them, we almost automatically solve for the in-betweens.

There is a contrasting effect of taking into account the opposite extremes that helps us clarify the problem and allows us to see functional solutions easier.

## Abstract the task

***In order to find effective solutions one must abstract the task. It must work on an abstract level in order to function on a concrete level. When you have found solutions to the primary properties, you add secondary properties (other theories prescribe long listings of ranked requirements before start, which will paralyze activities). Create different concept solutions by using your own creativity first, followed by creative dialog with others.***

From a more philosophical viewpoint, you can see all concrete things as connected to the natural laws by a hierarchical structure of concepts. That structure is correct, or true, for all concrete things, systems, etc, that function as they should. Now, when trying to design a product, you will find it much easier if you see the conceptual structure of abstraction "above" the product you intend to design and work yourself up and down that structure.

If you run into problems, move upwards to a level that works and then down another leg. Figure 13 is an illustrative example of a tree structure of the kind that is useful in concept development.

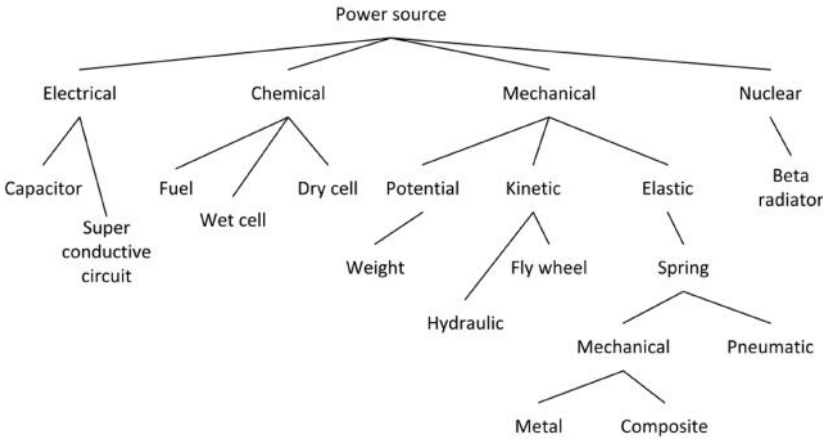


Figure 11. A tree structure showing different abstraction levels

It is often wise to create a tree structure on a piece of paper of the concepts and other factors. This is akin to making so-called mind maps—a useful creativity tool.

## Use BAD–PAD–MAD

***To move downwards from the top abstract level, you perform Brain Aided Design (BAD), an intense mental activity, paired with sketching, Pencil Aided Design (PAD), and/or the development of simple models, Model Aided Design (MAD), for example made of clay, cardboard, wood, etc. Gradually you move from generating non-dimensional solutions to increasingly well-defined solutions. Only when it is time to optimize the design do we start using to the computer (CAD = computer aided design). Other theories, e.g. Concurrent Engineering (CE) prescribe the use of computer tools (CAX) from the start. This has proven to be inefficient and harmful to creativity.***

**Brain Aided Design (BAD.** Start by thinking! A large part of concept development and engineering design is the joggling of objects in the mind. Design engineers visualize the design and try out different solutions in their mind. It is therefore important to train this ability, and to learn to create the right circumstances, necessary for the ability to function at optimum level.

**Pencil Aided Design (PAD.** Pencil and paper are the most important tools for concept development and for solving problems in engineering design. By drawing on paper there is created a direct link between the thoughts in the brain and the visual impression from the picture being drawn. The kinesthetic sense, muscle memory and hand-eye coordination is activated.

The hand motion is important for brain activity when looking for a solution, especially creative ones. Many images are created that are simultaneously viewed and processed by the mind. The paper serves both as memory and test bed for new solutions.

**Model Aided Design (MAD**. It is often useful to build and test some simple models to verify the function of concepts or to increase ones understanding of the concept. Favored materials are clay, Styrofoam, LEGO, balsa wood, and cardboard.

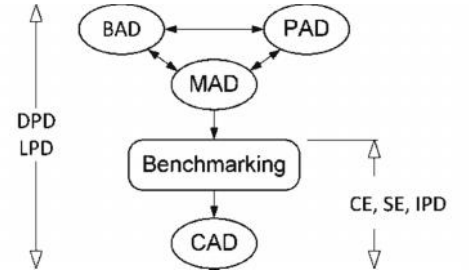


Figure 12. Comparison with CE, SE, and IPD

## Reinvent the wheel!

***The intent of this rule is to emphasize, in a striking way, the importance of not habitually reuse old solutions and to be creative yourself first, before you study the solutions of others. For, if we begin by studying the solutions of others, our creative ability will be blocked for a long time. In comparison, your own sketches on paper will always lose against the ready-made solutions that are already in production. To start with benchmarking, as some methods prescribe, is thus completely wrong.***

## Use creative dialog often

***It is very important that there is an open dialogue between all participants in a product development project. Research has shown that "creative dialogue" is the creativity method preferred by experts.***

***There is often a lively dialogue during coffee breaks. These are not wasteful, but are moments of creativity in progressive teams.***

Notice the difference between dialog and discussion. Contrary to dialogs, there often are in discussions and formal meetings one or several locked-up standpoints that are defended in a debate-like form. To foster a true dialogue between team members is important in all dynamic development projects.

The method is very simple. Two, three, up to four people sit together around a table (or stand at a whiteboard) with lots of paper and at least one pen each.

One person describes his idea, as he sketches on paper in front of him. In the meantime, the others try to understand the idea and try to come up with their own ideas of how to improve on and help the described idea to succeed.

When the first person has exhausted his supply of ideas, the next take over, and so on, until you feel that the possibilities of the first described idea has been exhausted. Then you make a summary sketch on an A4 or A3 and describe the idea in a few words. When finished, you fasten the paper to the wall.

The next idea is brought up and the process repeats.

## Identify primary and secondary demands

***When the primary functions and characteristics are determined and we have design concepts/solutions for these, it is important to find design solutions for secondary properties and functions. They are, for example, service-friendly solutions, solutions that facilitate assembly and disassembly, recycling, etc.***

It is important to study various user categories (primary and secondary users, service personnel, etc) to see how they actually use the product, and to get impulses to continued future development. First, you should become user yourself and try the product. Next, you should observe and talk to users. It is often said that, by talking to and observing at least eight users you capture a majority of the product's problems.

It is important not to try to solve all problems at once, but to start with the primary functions and when they are resolved, attack the secondary functions.