Information Knowledge Systems Management 10 (2011) 151–173
DOI 10.3233/IKS-2012-0191
IOS Press

Chapter 9

Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning

Larry J. Leifer* and Martin Steinert

Abstract: Over the past thirty years, a powerful methodology for innovation has emerged from engineering and design thinkers in Silicon Valley. It integrates human, business and technical factors in problem forming, solving and design: "Design Thinking." This human-centric methodology integrates expertise from design, social sciences, business and engineering. It is best implemented by high performance project teams applying diverse points-of-view simultaneously. It creates a vibrant interaction environment that promotes iterative learning cycles driven by rapid conceptual prototyping. The methodology has proven successful in the creation of innovative products, systems, and services.

Through courting ambiguity, we can let invention happen even if we cannot make it happen. We can nurture a corpus of behaviors that increase the probability of finding a path to innovation in the face of uncertainty. Emphasis is placed on balance of the questions we ask, and the decisions made. A suite of application examples and research finding will be used to illustrate the concepts in principal and in action."

Keywords: Causality, corpus of behavior, corpus of knowledge, coordination, instrumentation, design learning, design thinking, engineering design research, global collaboration, performance measurement, teamwork

1. Why designing products and systems translate into changing human behavior

At first glance, the activities of designers and product developers may have only limited value when trying to understand and facilitate the causality of change. Let us assure you that this is not the case. Of course, design and product development are, on the surface, about creating tangible products and systems including complex software systems. Classically engineering design, our root discipline, aims to generate alterative solutions to satisfy performance requirements and software specifications. Testing critical functions is a central feature of the iterative prototype development process. We generally plan development in four distinct phases: 1) concept generation, 2) concept screening, 3) concept scoring and 4) concept testing (Ulrich & Eppinger, 2008). The result is a series of technical prototypes and test results upon which an internal or external client can make a production and deployment decision. Traditionally this approach is highly analytical in nature and driven by system decomposition with a focus on sub problem solutions, often underestimating the complexity of system integration.

Sub systems are typically optimized in isolation from other sub systems. Feedback loops, often mythical, throughout the process are expected to facilitate coordination and cross system integration. Iteration is expected to lead to convergence to a successful solution space. Unfortunately these formal models

*Corresponding author: Larry J. Leifer, Center for Design Research (CDR), Department of Mechanical Engineering, Stanford University, 424 Panama Mall, Stanford, CA 94305-2232, USA. E-mail: Larry.leifer@stanford.edu.

focus on knowledge as knowable, as truth, immutable in the context of changing system specifications and usage scenarios. The missing piece is that the model is a protocol for capturing, disseminating, and re-using tacit knowledge, and especially how to use it differently in the face of changing requirements (Nonaka & Takeuchi, 1995).

Over the past two decades, this largely linear and sequential water-fall model has given way to the spiral like notion of rapid conceptual prototyping and triple-loop-learning. Consulting companies like IDEO and Frog Design have achieved notable success in a wide variety of industries through the use of adaptive design thinking and semi-formal use of a "coaching" model that has some members of each development team explicitly focused on the team's behavior pattern with an eye to focusing activity on the critical tasks from a system integration point of view. A parallel movement in the software industry operates under the tag line, "agile systems development" and our favorite protocol "scrum development" (MacCormack, 2001).

While still grounded solidly in engineering principles and construction, we have realized that the Meta level issue of customer adoption should be the defining parameter for measuring the success or failure of a new product or system. Changing the actual behavior of the user in positive ways, whoever that may be, determines success (we will take a closer look into the specific problem of identifying the user(s) later, it is not as obvious as it may seem).

A growing number of practitioners, researchers and teachers are combining novel dimensions of product design and development into a rather loosely labeled box called design thinking. In it, one can nowadays find tools to, amongst others:

- Focus on human values
- Discover customer needs (need finding)
- Employ rapid iterative learning cycles (prototype)
- Apply human factors and ergonomics
- Be mindful of process (design team behavior)
- Manage design team composition and dynamics
- Use product based learning paradigms
- Show, don't tell
- Prototype (build) to learn
- Design the customer experience
- Maintain a bias towards action (observe, prototype, test)
- Give questioning an equal or greater status than deciding

All of these rather recently added design and development process aspects aim at enhancing the chances for the product being adopted. They also aim to increase the utility in relation to the behavioral change required by adopting the new solution. This points to the fundamental message we believe we can contribute to this research on the causality of change. The successful introduction of any new product, service, system, organization or process requires the solution to overcome the inherent behavioral barriers to change. Barriers might arise on behalf of sometimes unexpected parties, stakeholders or system advocates that have a priori not been identified as critical to the particular new design. The cost of behavior change can define market success. See Aquino, Steinert, & Leifer (2011) for an example in the medical device industry.

In this chapter we will present some of our tools and examples for causal behavior change that enable successful adoption of a new designs and system paradigms. We believe that these tools and "bodies-of-behavior" (physical activities and methodologies) may serve as an analogy for other kinds of complex system challenges.

A word of caution on our usage of the word "new" and on our research and teaching focus – Coming from Stanford and Silicon Valley, people expect us, and we attempt to deliver radical or transformative new solutions and designs that obsolete exiting barriers and problems (while possibly creating new ones). We are not usually engaged in incremental innovation or improvements.

1.1. Translating the change problem

As described in Chapter 1, the grand challenges of this book address complex social and technical systems and the causality of change. Depending on each discipline, these concepts carry different meaning. To put key terms into perspective for the reader, we would like to translate the challenge space into a design specific language and wording.

A good starting point for any new design or product development cycle is a problem formulation or design challenge as we call it. For example, "Redesign (read re-invent) the driver-car interaction." Due to our focusing on radical new solutions, we usually like to tackle customer pull challenges of a complex nature that are based positioned in the future (3–5, sometimes up to 20 years). That allows us to truly understand and (re – define) the problem, a prerequisite to ideation and a stimulus to interesting new solution concepts.

Technology push scenarios, ready solutions such as new materials with hitherto unseen attributes that are still looking for an appropriate problem or application area are a much tougher nut to crack. For example, what application domain will provide graphene[1] with its first commercial application?

In order to address socio technical complex systems and causality of change from an engineering designer's perspective, we have translate this book's challenge as follows:

1.1.1. Technical

As described in the beginning of this chapter, the origins of product development and design lie in classical engineering, mechanical and electronic, all solidly grounded in classical Newtonian physics. Thus, by definition, all products that engineers develop are fundamentally of technical nature.

1.1.2. Systems

No physical product can be designed without taking its context into account. The user has to act, more often today, interact within a certain contextual environment. The trend is beyond embedded electronics and mechatronics towards connected smart products. As most design challenges require more than a stand-alone mechanical solution, most of today's products should be more accurately described as systems. Due to the inherent system integration problem, it is necessary to establish diverse, pan-disciplinary product development teams.

1.1.3. Social

All products or systems interact or influence directly or indirectly certain user and stakeholder groups. Thus, essentially, all products and systems aim to facilitate a certain user adoption and behavioral changes. The social acceptance of new solutions is also essential. One must design for the social context and the product itself must behavior in socially expected (acceptable) ways. Engineering design has therefore integrated user centric design and need finding. It is even borrowing from cultural and physical anthropology in its quest to understand the usage context.

[1] The Nobel Prize in Physics for 2010 was awarded to Andre Geim and Konstantin Novoselov for groundbreaking experiments regarding the two-dimensional material grapheme. Graphene is a flat monolayer of carbon atoms tightly packed into a two-dimensional (2D) honeycomb lattice, and is a basic building block for graphitic materials of all other dimensionalities. It can be wrapped up into 0D fullerenes, rolled into 1D nanotubes or stacked into 3D graphite (Geim & Novoselov, 2007).

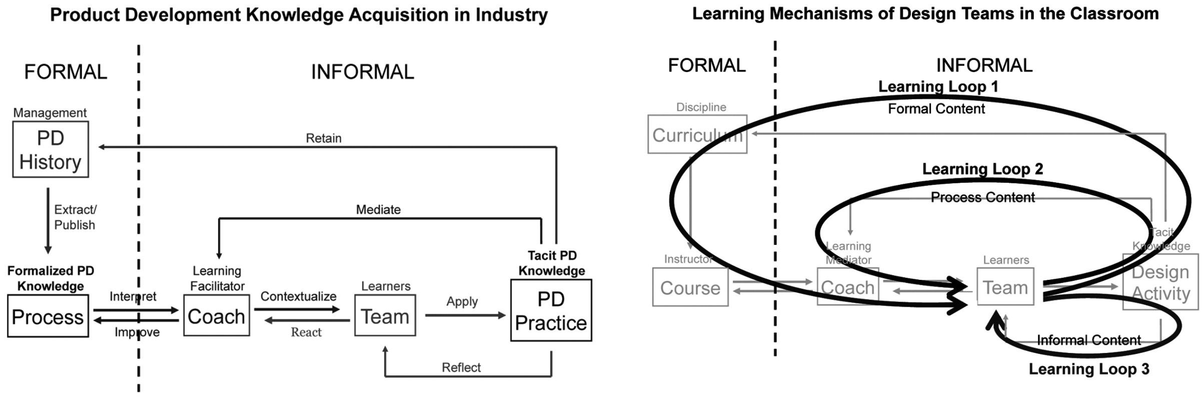


Fig. 1. A product development knowledge model based (Eris & L. Leifer, 2003).

1.1.4. Complex

The combination of the technical, social and systems dimension provides for a vast and ambiguous solution space. Radical new designs may completely solve the existing challenge by redefining existing solutions. Of course they may also create an entirely new set of problems and challenges (to be addressed with the next generation). When we are trying to understand the initial assumptions and boundary conditions as well as the interface, integration and adoption issues, not to speak of the implications of radical new solutions to existing challenges, we realize that we are almost always dealing within complex systems.

1.1.5. Change

Success ultimately depends on the willing adoption of the new system by the user. This translates into the solution's capability to overcome resistance to change. Though, as a discipline, engineering design may not contribute to the fundamental understanding of the causality of change, we belief, that our grand task of creating better designers and a better design process produces better products and services and increases the chances of success for change. Change does not come naturally to the majority of people, engineers and users alike. That is why we attempt to understand the underlying principles and why we are developing techniques to overcome barriers to change.

In essence a product developer and designer is constantly struggling to improve the outcome of a complex socio – technical system interaction, as well as the project organization and the work process of a development team.

1.2. A product development knowledge model

In order to structure possible insights that we have gained on change, let us introduce a product development knowledge model. Initially based on empirical data from a large US auto producer (Eris & L. Leifer, 2003) and later tried and tested in our ME310 teaching environment we are differentiating between three simultaneously occurring loops of knowledge acquisition or learning. ME310 is the a three quarter project based mechanical engineering graduate course at Stanford University that teaches 10 global teams based on a real industry challenge.

Learning or knowledge acquisition is a prerequisite for change, whether on a design or on an organizational level. Based on this product development knowledge model, we have identified three learning loops.

Learning Loop One is based on explicit knowledge. It stretches beyond the informal product development team into the formal organizational structures. Due to its explicit nature, it can be collected and managed and synthesized into formal processes. It is mostly comprised of quantitative technical data such as business processes (BP), computer aided designs (CAD) files and workflows, data warehouses, algorithms, repositories etc. Learning in loop one is mostly aimed at retaining project knowledge (facts, syntax, what, and how). It may comprise analytical activities and tools such as databases and may involve simulations. Learning loop one usually allows for optimization and incremental change. However, due to its institutional character the change process is relatively slow. Section 2 addresses the change relevant aspects of loop one. The vast majority of academic and business literature lies in the conduct and architecture of Loop-1. (Argyris & Schon, 1978).

Learning in *Loop Two* takes place within the informal space of the product development team and comprises the informal process content. Learning occurs during the exchange between the product development team members and the team's coach(s). The coaching role is often tacit itself. Coaches act as facilitators between the formal organization, its formal protocols, and actual team processes (typically unspoken, undocumented, profoundly tacit). Loop two is based on concepts, semantics, and architecture and asks questions like when and why. It is the locus of application for many of the design-practices that allow faster learning and better output, to be explained in Section 3.

The least studied *Learning Loop Three* concerns tacit knowledge embedded in the teams themselves and the established practices. Team members learn from each other and prior team's experiences by applying, reflection upon and improving informal practices. Changes stimulated through learning in this third loop are fast paced. It forms the rational that explains the success for activities such as fast iterative cycles and rapid prototyping. In Section 4 we will present some of the research findings that shed light onto the underlying principles of design-thinking based activities.

We do not explicitly aim to understand the causality of change in design research. However, the process, practices and environmental setting of creating substantial change by means of new products or services, ultimately translates into changing behavior. Internally, we explicitly change the behavior of designers with the intent to augment their ability to change user behavior. How to achieve these changes is at the core of our research and design activities. We believe that our insights into how to initiate, support, and facilitate the creation of radically innovative new products and system designs may help to understand how to trigger and sustain change in complex socio-technical systems at large.

2. Formal institutional and procedural learning (Loop 1)

The institutional challenge begins with capturing what is known. Procedural implementations are best suited to capturing the facts. They are least successful at capturing the behavior, how knowledge was used or ignored. Institutional efforts to extend knowledge capture to the tacit level of design-development team activity are largely unknown. One exception to this rule lies in the design thinking research program at Stanford University wherein student design development teams are accessible and open to observation within the limits of human-subject study protocol. Many of the research finding we are about to share come from this organizational behavior "flight simulator." There is a substantial body of literature that address issues ranging from project team size and setup (self organizing open source projects vs. centralized R&D team) to supportive tools (knowledge management systems, wikis, computer supported cooperative work (CSCW) tools etc.) and supposedly innovation enhancing processes (stage gate model, lean approaches, design for X).

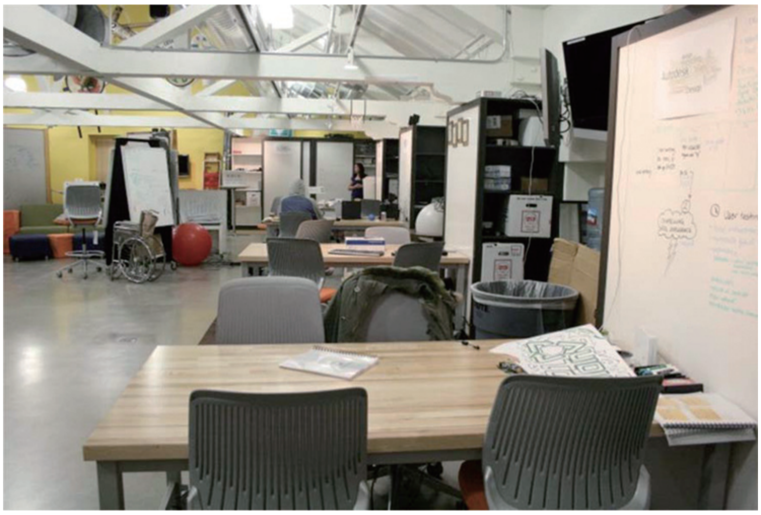


Fig. 2. Creative collaboration spaces are open, flexible, and adaptive with ample white board space and an informal atmosphere that features past designs hanging in the rafters.

Based on our particular experience, any one specific design-development organizational recipe can work for incremental innovation but most are not appropriate for enhancing disruptive product and system development. For this overview on Learning Loop One, we would like to focus on the factors that seem to influence the ideation and creative energy and output the most: physical space, the absence of fixed processes, and an overarching institutional practice of letting change happen. This last point stands in sharp contrast to trying to make (manage) change happen.

2.1. Space

Space has emerged as a key factor to facilitate change. Through adapting the physical environment, organizations are able to lower hierarchical boundaries, enhance ideation and creativity, foster and accelerate prototyping and generally increase the rate of learning and change. The key concept for the spatial setup is flexibility (adaptive/agile work places). Space ought to allow for and support any kind of ideation and prototyping activities. As will be described in Section 3, going through a number of rapid iterations, testing ideas and the boundaries of the solution space via prototypes, allows the project teams to increase their rate of learning significantly. Founded in 2004 by David Kelley, Professor at Mechanical Engineering and co-founder of IDEO Product Development, the Hasso Plattner Institute of Design at Stanford, more commonly referred to and globally famous as the d.school, is continuing to shape and incorporate lessons learned. Being in its fifth building in six years, continual changes and experimentation with space and furniture alike has become the normal modus operandi. The key concepts include:

- Use flexible room separators instead of fixed walls move them daily as needed, or just maybe to stir things up

Aim to obtain the largest possible contiguously open floor space. It is easy to temporally separate it again based on the project context, not vice versa. As a result, if and when we use room separators, these are very flexible and rather light in nature. Indeed all walls are in fact simple and rather cheap whiteboards

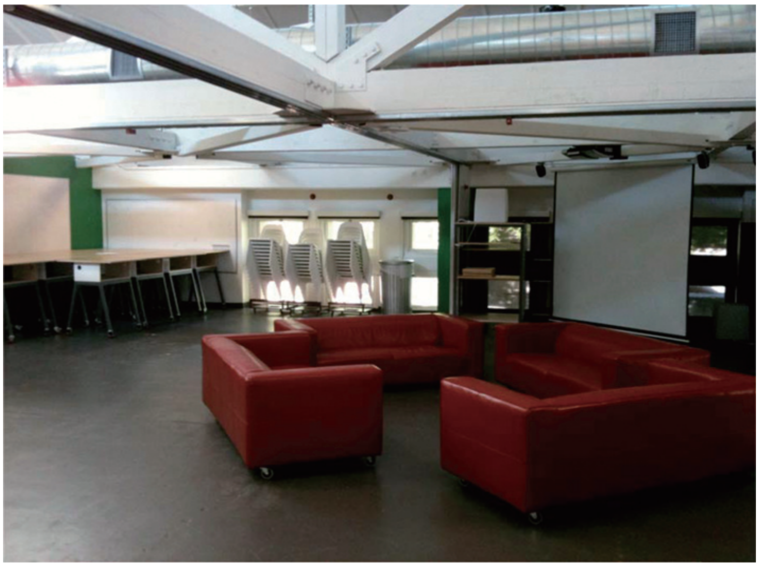


Fig. 3. d.school studio (aka classroom) in its standard starting configuration. There are no rules, only to put it back into this configuration after the classes. (photo: Everything got to be on wheels by Ewan Macintosh, 2010).

- All furniture is easily movable and modular to serve multiple, often previously unexpected purposes

The next picture shows the standard configuration of a classroom, called studio in the d.school language. Though it can support traditional teaching activities, it allows for almost any other standard setup and also supports hybrid activities such as workshops, prototyping experiences, round tables and in our case, red sofas.

Supported by state-of-the-art electronics, these rooms are truly multifunctional and agile. They support the instant design of new configurations without prior planning.

- All furniture has evolved with a focus on enhancing creativity and lower barriers to ideation

The key idea behind all furniture is flexibility and collaboration. The table depicted in the next picture for example is in its third iteration.

Its size is deliberately too small for four people, enhancing the chances that the team will incorporate the abundantly available whiteboards.

- One of our more famous student quotes goes like this . . . “ah, I get it, if you give me and 8.5” x 11” sheet of paper to “think” then I can only get ideas that big . . . [but] . . . if you give my a 4’ x 8’ sheet of white board I can have ideas 30–40 times larger (smile)”

The table and its chairs have been increased in height, in order to minimize the difference in height, which translates into hierarchical differences when working with one or more team member standing at a whiteboard. A rolling base and additional storage space complements this setup.

- Workshop facilities “lite” are easily accessible and incorporated in the normal work and teaching space promoting and supporting rapid conceptual prototyping

As a general rule, we aim to minimize the time and barriers to learning, which is to prototype. Therefore we attempt to integrate workbenches and workshop facilities directly into the project team space. The

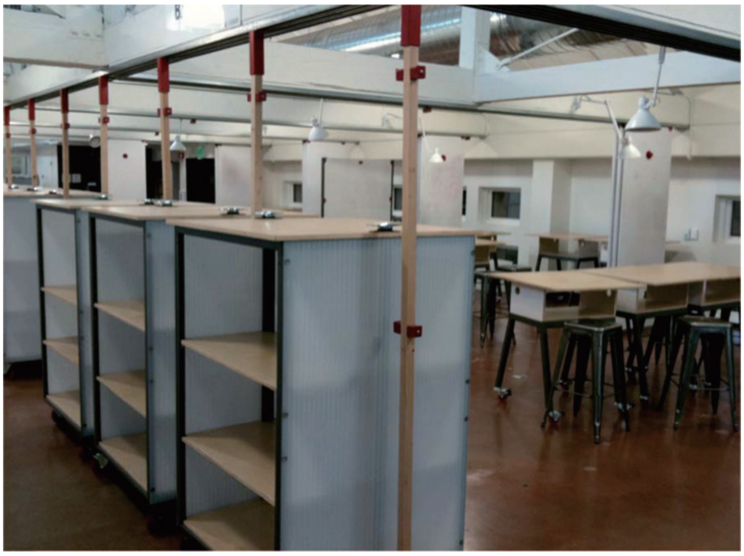


Fig. 4. d.school student work area, not the combination of tables and seats and whiteboards that allow the instant creation and redesign of ideation zones (photo: Movable feast of furniture & storage by Ewan Macintosh, 2010).

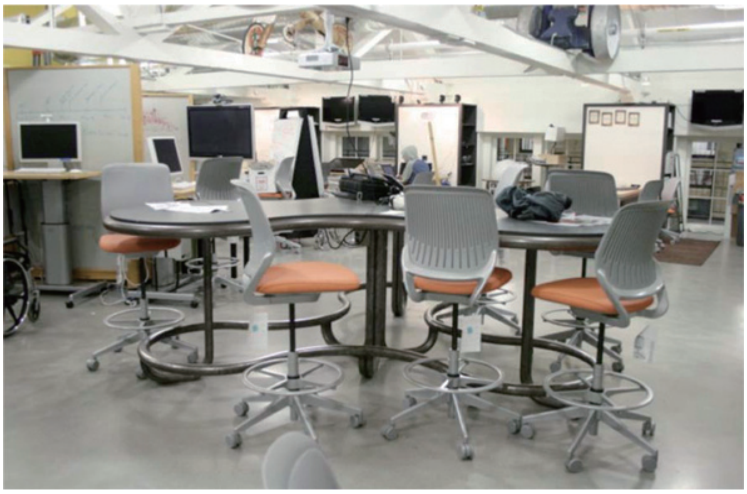


Fig. 5. ME310-Global Studio: 10 design-development teams work here with 24 hour a day, 7 day a week access. The screens you see are primarily used for video-conferencing and document sharing with global partner teams (T. Carleton & L. Leifer, 2009).

loft of ME310. (ME310, 2011), the final project based learning course in mechanical engineering at Stanford University, leads this concept.

Materials, a solid workbench and power tools for rapid prototyping as well as mechatronics are available in the middle off the team space. In fact, together with the coffee machine, fridge and microwave this ensemble forms the "energetic" heart of the space, positively influencing all teams working around it. Additionally, students have access to the full Stanford University machine shop, the Product Realization Laboratory (PRL) (PRL, 2011), which is situated less than 100 meters away. It supports design, creation and innovation through traditional machining, woodworking, foundry, plastics molding, welding, finishing. State-of-the-art computer-aided drawing, manufacturing, and prototyping systems are also available.

Our lessons learned:

- Avoid expensive solutions that bind infrastructure investment; instead use minimum commitment prototypes to facilitate rapid redesign/learning
- Building furniture and support infrastructure in-house as needed by the project teams.

These last two key concepts further support the notion of flexibility and continual change. We believe that we cannot correctly foretell the usage scenarios and therefore we do not want to preplan in detail the space requirements. Instead we focus on maximizing flexibility while minimizing financial requirements. We are using standard modules repurposed from other products and good but not high-end materials in order to minimize costs. The idea is to have only very limited equity tied into infrastructure that only depreciates over years. Instead we know that we are going to redesign the space, furniture and equipment constantly, we know that we learn and change constantly. This allows the project teams to redesign their space and infrastructure as needed. It allows teams to go on pathways not planned for, the ultimate aims still being the generation of great new product and system ideas and concepts.

2.2. Absence of fixed processes

After having experienced many cases (and some very few studies) that point to the inability of any particular fixed process model to support the output of radical new products and services, our research turned to DARPA in order to better understand how to support change. DARPA, the US Defense Advanced Research Projects Agency has supported groundbreaking projects ever since its establishment in 1958. Contrary to the majority of other business, government, or academic research organizations, DARPA is specifically and solely focusing on the creating of radical engineering and system innovations. Their mission is to "maintain the technological superiority of the U.S. military and prevent technological surprise from harming our national security by sponsoring revolutionary, high-payoff research bridging the gap between fundamental discoveries and their military use" (DARPA, 2010) In fact, all projects may be defined by a set of criteria that have found their way into the urban directory as "DARPA hard" (Urban Dictionary, 2011) (Van Atta et al., 2003). All projects must be

a) Technically challenging (beyond current limits),
b) Actionable (proof of concept or prototype),
c) Multidisciplinary (complex) and
d) Far reaching (advances on a grand scale, radical).

Tamara L. Carleton from the Center of Design Research at Stanford University, who graduated in 2010, has securitized this remarkable test bed in her doctoral thesis (T.L. Carleton, 2011). The most interesting observations, the ones that also concern change, are the notion of projects and people being vision driven, the focus on workshops and prototyping, the absence of a formal process documentations or project management rules and, last but not least, the leadership driven decision model that does not rely on peer review or communal decision building mechanisms.

The selection of DARPA programs depends on the creation of a vision. It is the starting point for any program, and the project champion, the program manager, embodies it. The ideation and iteration of this vision serves as the central focal point for the usually dispersed sub projects, teams and stakeholders. Envisioning a certain technological future does not define or limit the future projects; it rather serves as indictors of the current direction of the organization's efforts.

Interestingly, the main instruments to generate, iterate and re-formulate such a vision are workshops and the creation of proof of concepts or prototypes at various stages. The first allows the socialization and evolution of the visionary ideas amongst all participating stakeholders, while the latter allows tangibly communicating and even testing the vision at various critical junctures.

The program and project managers also enjoy a remarkable freedom from established processes and rules. No established system or documentation requirement is forced onto his or her activities. Prototyping is the norm and the specific activities follow the actual demand of the specific task at hand. No institutional models are forcing people and their behavior into fixed corsets for the sake of generating economies of scale. Innovation and change is the generation of the new – the primary goal is the best outcome at certain budget constraints, not it's process efficiency in terms of minimal resource allocation.

Another point to consider is the ways that go and no-go decisions are prepared and executed. Instead of relying on a peer review processes or committees or other group-based decision tools (not to speak of pseudo quantitative stage gate filter variables) decisions are taken by the leaders who ultimately bear the responsibility. Failure is accepted and preplanned. The underlying rationale is that peer review and committees are in fact hedging mechanism for taking tough decisions at the extreme end of the possible solution spectrum. They will inherently favor outcomes close to the sample's median opinion. Hence, traditional decision tools would prevent DARPA from actually attempting to deliver radical innovations. All of these activities, and most importantly the absence of fixed processes, serve to generate change that complies with the idea of DARPA hard, as described above.

2.3. Ability to let change occur rather than manage it

As the prior discussion on space and flexibility as well as on the absence of fixed processes indicated, the major concept to support change and learning, the generation of radical new product and system solutions, is to allow change to happen. We do not assume to have control over the existing solution space, so we cannot preplan or manage it. Indeed, we are aware that the concepts that challenge the established dogmas have a higher chance to deliver radical improved value. Any systematic and fixed support system, inhibiting the creative use of space and the employing and combining of new processes seems to counter the notion of change. Therefore we attempt to provide the physical, organizational, procedural and mental environment that allows the project teams to experiment and to prototype. This becomes especially difficult when proposed solutions counter the experiences and knowledge models of the professors and coaches. Instead of prematurely ending the iteration processes at this point, we allow, indeed support, the testing of theses ideas and concepts. Very often, a failed prototype test, the hitting of the boundary of the possible solution space, generates the winning insights for either an extreme solution along that line or, even better, a new way that allows circumventing the existing limitations. To generate this kind of change, we attempt to minimize institutional, organizational and procedural boundaries. We emphasize and support flexibility, and we force ourselves to let change happen. Hence we do not prescribe procedural recipes. We rather teach skills, moves and demand tangible prototypes that requirement or specification lists. A word of caution from coach to coach, allowing and even fostering for this kind of ambiguity is difficult and demanding for the coaches and requires a conscious effort especially on behalf of individuals who have to unlearn their organization skills to a certain extend.

3. Design process learnings (Loop 2)

Almost all of our activities aim at accelerating ideation and creation; in other words we aim to accelerate learning. One of our dogmas for our design process, you might also call them design rules, is to initiate

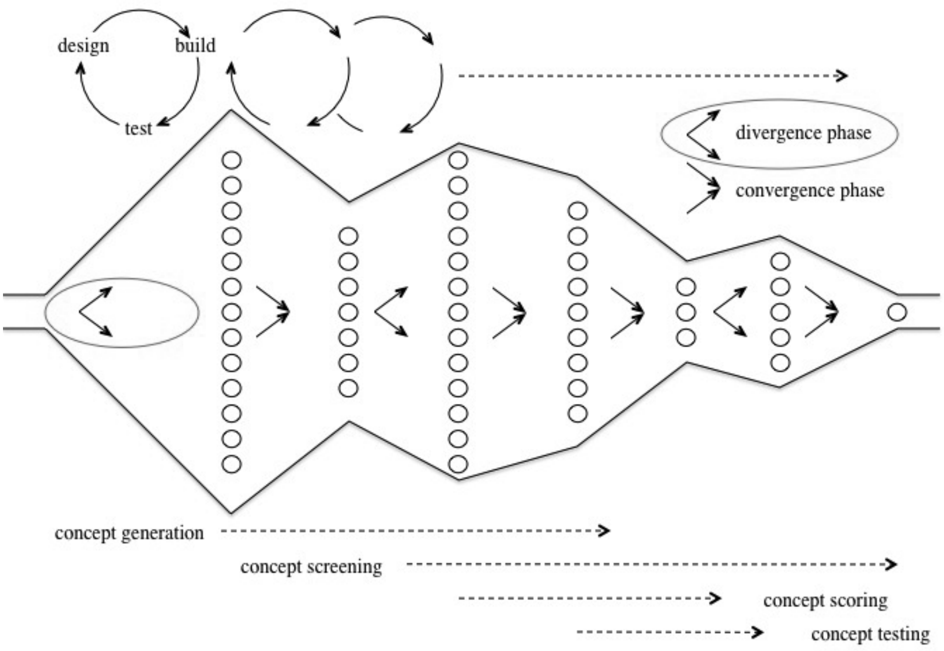


Fig. 6. Design process as iteration of divergence and convergence steps or prototype cycles, adapted from [Steinert, Leifer under review].

very rapid instances of change. Change, embodied in tangible prototypes, can be tested on or against user behavior. To foster the cross-team ideation activity, we deploy collaborative tools and create a physical space that encourages an even faster rate of learning or change.

3.1. Design process

Contrary to the classical and rather analytical design process applied for the development of incremental changes, the design process aiming at radical changes can be seen as an iteration of divergent and convergent activities (Alexander, 1964). Banathy describes the divergent activity as "...consider a number of inquiry boundaries, a number of major design options, and sets of core values and core ideas. Then we converge, as we make choices and create an image of the future system" (Banathy, 1996, p. 74). This divergent – convergent process may be depicted as slowly closing funnel, linear over time (Cross, 2000) (Ulrich & Eppinger, 2008), or as repeating design cycles, spiral like, that iterate through the generic prototyping phases of design, build and test (Thomke, Fujimoto, & Research, 2000).

The classical convergent phase is about optimizing the answer; it is deductive and inductive in nature and may comprise simple tools such as the Pugh Chart (Pugh, 1996) or a Quality Function Deployment (Hauser & Clausing, 1988) for example, but it may also run on complex model simulations and optimizations.

The design thinking approach that we are favoring not only emphasizes the circular or spiral nature of the process (feedback loops were common but limited in the classical process models), but it clearly identifies the need of divergent search activities. Developers are constantly and rapidly going through design-build-test cycles. In each cycle, during the divergent phases, we are focusing on the problem rather than on the solution, trying to understand who really is the user, which elements are truly involved, how many other ways are there to solve the problem, can we rephrase the challenge, can we circumvent the problem? We are generating concepts. These divergent activities usually result in a number of ideas or concepts that are in a next step built and then down-selected by testing. The underlying principles of Generative Design Questions (GDQs) and Deep Reasoning Questions (DRQs) will be explained in Section 4.

3.2. Rapid and tangible prototyping

In our design process we concentrate on creating prototypes as fast as possible in order to be able to test particular ideas, the design hypothesis behind the prototype. Speed, the acceleration of learning is key. As a result, our prototypes tend to be of low resolution and physical or tangible rather than virtual. Depending on the design stage, whiteboard, simple cardboard and duct tape constructions, prototypes made from wood or clay etc. might be created.

Each prototype is built to test a specific idea and/or a system interaction. They range from simplistic rough artifacts that merely resemble an idea (communication prototype), to lookalike prototypes (conveying certain external property ideas) to critical functional and functional prototypes (technical proof of concepts), to alpha and beta prototypes. It must be noted that later stage prototypes cost an order of magnitude more in resources, both in time and money than early prototypes. It is therefore essential to concentrate on the early stage or fuzzy front end of the new product design. One of our recent yet key insights, based on dissertational related work by Jonathan Edelman (J. A. Edelman, Banerjee, Jung, Sonalkar, & Lande, 2009), is that the choice of the prototype material or environment, directly influences the amount and degree of the generated alternatives.

The breadth and depth of the solution space explored seems to relate to the sophistication or resolution of the prototyping materials employed. A sophisticated CAD prototype is least likely to be considerably changed in following iteration cycles. The product architecture is implicitly fixed and the software and its capability limits possible ideation changes.

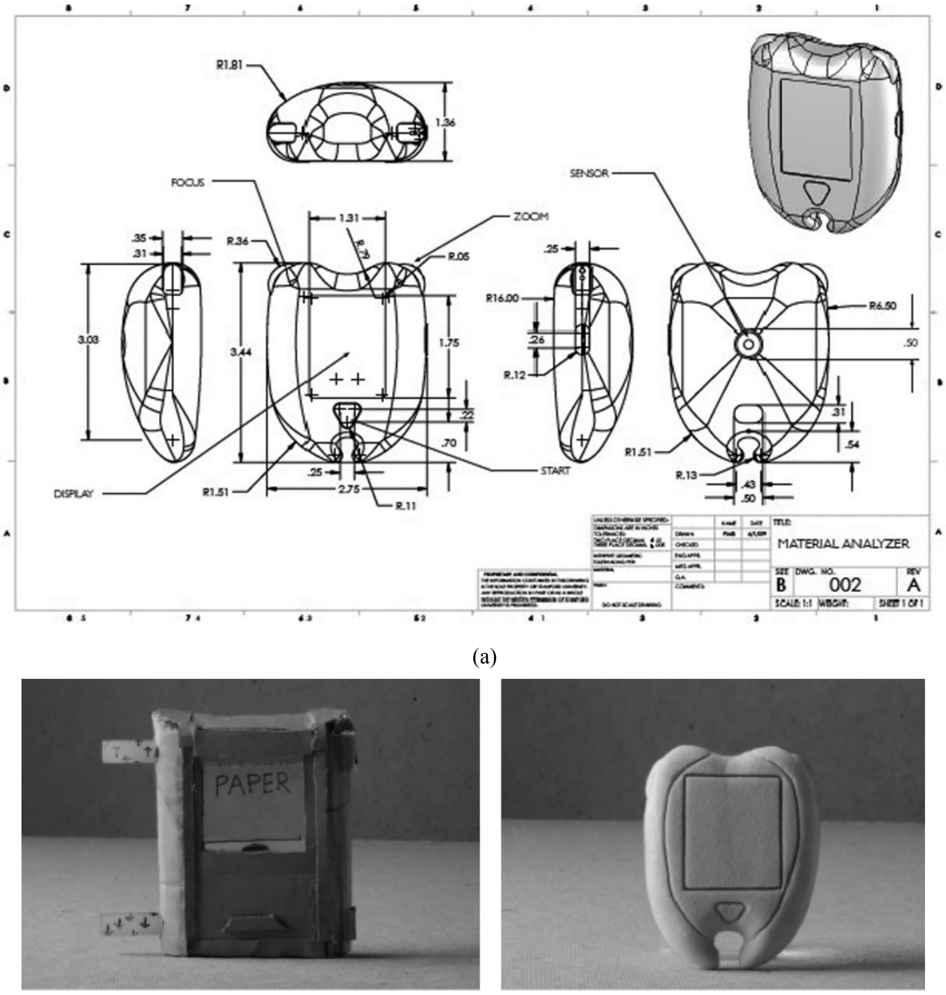
Tangible 3D prototypes allow the creation of more alternatives with relative ease. These types of lookalike prototypes are especially good in conveying ideas and form factors to non-specialist users. However, as can be intuitively seen by the foam model example depictured above, once this level of resolution has been reached, changes tend to be rather incremental.

If we contrast this to using very basic prototype material, simple cardboard or even just a sketch (please have the proverbial napkin at Buck's in Woodside in mind[2]) the possibility for more radical and faster iterations and thus learning is obvious. As a rule of thumb, the early stage product development determines the level of radicalness of the final solutions. We therefore advise product development teams to stay in this early phase for more than a third of the entire available project time. We have to force ourselves to abstain from entering solution optimization in order to gain intimate awareness of the problem space. This increases the chances for us to generate the real breakthrough idea we are looking for.

3.3. Need finding, body storming, user experience enactment

Whereas the late stage development is focused on optimization of performance, cost, and reliability, the early phases of the product design process require a different set to tools and behaviors.

[2] "... the famous Buck's of Woodside restaurant, which is well known among Silicon Valley entrepreneurs as the location where many VC investment deals have been signed" Wikipedia 2010, Woodside.



(a)

(b)

Fig. 7. (a) Details specifications and computer-modeled prototypes inhibit ideation. (J. Edelman & Currano, 2011). (b) Rough, low resolution, prototypes facilitate ideation (J. Edelman & Currano, 2011).

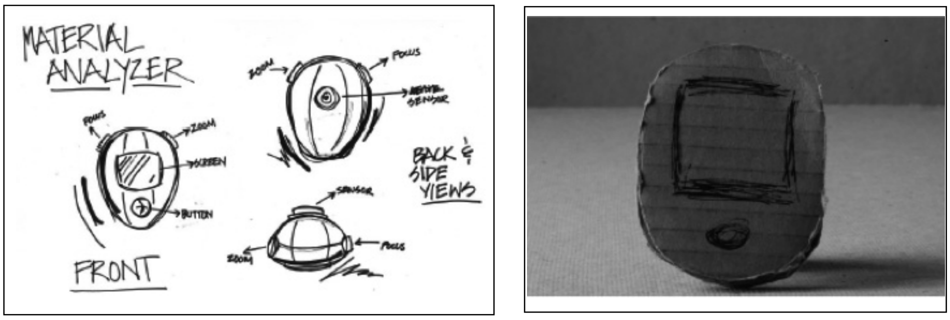


Fig. 8. Tangible 3D prototypes, right, facilitate associative memory, analog thinking, and exploration better than 2D sketches (J. Edelman & Currano, 2011).

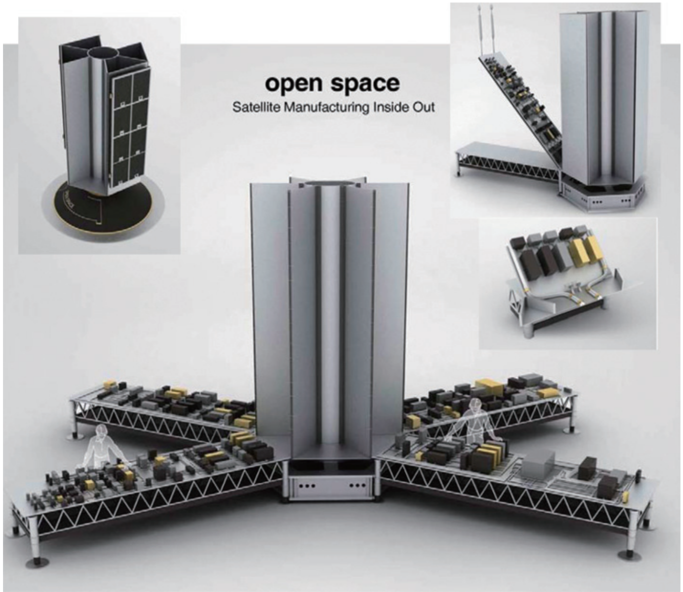


Fig. 9. Communication satellite open space architecture to tackle the common (75%) late box replacement issue. Note the flower inspired panel setup around a core structure. Heat pipes regulate temperature (*ME310 EXPE 2011*, 2011).

3.3.1. Need finding

Central at the early stage of the new product or system concept design is an intimate understanding of firstly, who actually is the user, and secondly, what are the real user's needs that we aim to satisfy with the solution? Often, projects start with a fixed set of specifications and requirements. This approach, very suitable for incremental change and innovation, focuses the attention and resources onto the optimization and execution of the selected concept. Time and again though, final solutions do not meet actual user needs and need to be re-designed before deployment can succeed. This costs significantly more money and prestige than conducting more exploration early on. Therefore we are especially concentrating on the first phase of the design process.

The first challenge lies in identifying the actual user to design for. Some iterations and perspective changes may unearth surprising users. Challenged to redesign satellite architecture, an ME310 team is currently focusing on the testing engineer as target user. The pre-launch testing process ties significant resources due to the fact that satellites have not been engineered for modularity/mass customization and access to the sub systems that have to be tested again is typically outside the satellite design team's concern. Testing and validation become a large fraction of system integration costs that are, in turn, a major factor in net deployment cost.

In a thoroughly unrelated case study, scrutinizing medical device development, it is not the patient, the obvious user, and his needs that are central for the success of a new product. Though any new solution must at least be equal in terms of patient value added, the real litmus test lies in the value gained by the hospital and insurance companies, in relation to the change required by the practicing medical doctors. Who is the user for whom we have to design for in this case (Aquino et al., 2011)?

Once we identified a single target user or a user system, we attempt to gather information on the underlying needs that ought to be satisfied by the new solution. While surveying and interviewing users does give valuable information, very often, users are themselves not capable of expressing their needs. Indeed when confronted with something absolutely new, for example a device based on a new technology or material, users can only draw from analogies and not answer from experience. Even if they can, very often their personal perspective is too limited to truly understand the problem. We find that observations, especially when analyzed systematically using video interaction analysis (we code videos frame by frame and quantitatively analyze the resulting process data (Tang & L. J. Leifer, 1991) result in a better understanding of the process and behavior we attempt to improve. To achieve the same, our community has been borrowing heavily from anthropologist. We attempt to embed into the problem, trying it out ourselves. As the literature of knowledge management tells, this direct tactile involvement with the problem is often the only way to transfer implicit procedural knowledge. As Nonaka and Takeuchi famously describe (Nonaka & Takeuchi, 1995), to build a home bread-baking automate that also kneads the dough, it was necessary to practice kneading with a baker. The development team would not have been able to uncover the complexity of the compress, pull, and twist action necessary to create dough that raises just right.

3.3.2. Bodystorming, user testing and user experience enactment

Bodystorming is the second important tool we employ. It is a design inspired technique that challenges the designer or user to imagine what it would be like if the product existed, and act as though it exists, ideally in the place it would be used. Rather than creating post-its and bullet point lists, we aim to engage an idea with improvised artifacts and physical activities to envision a possible solution. In fact we aim to employ bodystorming instead of using classical brainstorming. The underlying idea is to tangibly create ideas and translate them into super rough communication prototypes. Very often the act of creating such a prototype storm generates artifacts on which we can better experience, test and improve the proposed solution. Also, bodystorming helps to convey and iterate the concept idea amongst the team; it allows the team to use the artifact and to enact a usage scenario. Finally it allows to actually go to users and to get their direct feedback on possible solutions. Last but not least, the tradition of fast prototyping, immediately during or after user testing, allows redesigning the prototype on the fly. User testing, learning and iteration are thus being combined seamlessly into fast cycles of change.

4. Underlying design principles, lessons from learing (Loop 3)

In this section on learning loop 3, we would like to focus on the informal creation and transmission of explicit and implicit knowledge. Combined with insights and information from the other learning loops, this area forms the Center for Design Research (CDR) at Stanford University's core research agenda. We target designers and the design process. Our primary aim is to understand how designers and developers work, why some are more creative and some more analytical, how we can improve the team composition and interaction, how we can quantify and structure or better not structure the design process . . . – We believe our biggest opportunity for change lies in better education and in improving the support of designers and their design process including the contextual environment and support tools. On the specific topic of change, we would like to present some key insights that might contribute to the understanding, measuring and maybe even fostering of change, namely: importance of wording, of questions, of gestures and of emotions.

4.1. Noun phrases as change indicators?

The first key insight is based on extensive research conducted by Ade Mabogunje from 1993 to 2007 at CDR (Mabogunje, 1997). He analyzed design teams with a special focus on the usage of language amongst design teams in various settings and projects. The first series of experiments analyzed team interactions in a simulated teaching environment. The second series included the design problem context and introduced time development minimization as dependent variable. The subsequent experiments included a focus on the generation of alternative prototypes and also on a parallel design process. A common result amongst all experiments was the emergence of the creation of noun-phrases as interesting surrogate variable for describing the design process. In fact, based on design documentation analysis and in a setting that favors radical new solutions, the number of distinct noun phrases created correlates positively with performance. Other meta-document data such as total length or readability do not add insights. Mabogunje's research suggests that the creation of new words, noun phrases, is an indicator or maybe even a driver of creative activities inside the design team. On the one hand this relation might allow creating "speedometers" of change by counting distinct noun phrases in real time, on the other hand this might provide yet another lever to support ideation amongst the teams. By actively promoting and encouraging team members to strive for new wordings we might be able to enhance the chances to break out of the established solution space that only allows for incremental changes. Of course this relationship is sure to be context-, language-, and culture-dependent. Verb phrases also correlated positively with innovation outcome. Interestingly the new phrase amounts to a "re-representation" that, in agent based software systems is often taken as a measure of learning. Please note that we have only established the existence of a correlation, not a causal relation, existences and direction of the latter remains to be explored.

In sum, it may be noted that the creation of new language correlates with the degree of change achieved. Any change beyond the existing frameworks may thus have to be accompanied by new language.

4.2. Change, a question-driven process?

Besides noun phrases, questioning has a special place in understanding the fundamental design process. Based on Eris's research at CDR (Eris, 2003) (Eris, 2004), we have identified that design is in fact a question driven process. Eris identified and developed taxonomy of questions asked while teams of three to four designers were engaged designing a Lego-prototype. It differentiates between Deep Reasoning questions (DRQs) and Generative Design Questions (GDQs). The first reflect convergent, the latter divergent thinking. Based on this metric, a real time analysis of the design process is made possible.

Much like noun-phrases, on a combined level, questions may act as design performance metric. There is a general positive correlation between the numbers of questions asked during design activities and the project team performance.

Looking at the two established subcategories, GDQs are prevalent in the conceptualization phase whereas DRQs were mainly used to reduce the number of generated alternatives. Figure 9 depicts this central paradigm for radical design processes.

In sum, GDQs are essential for preserving ambiguity. They generate alternatives, reframe needs and drive the creative negotiations amongst the design team. DRQs are more prevalent during concept assessment and implementation, and generally aim to reduce the number of alternatives.

DRQs are based on tools and analytical skills that are rather classical and taught and implemented abundantly. GDQs however very often contradict an organization's natural tendency to control and manage. They are central for the creation of change as they allow the opening up of the solutions space for radical new ideas to emerge. Basing change mainly on DRQs will lead to incremental learning only, as no substantially new concepts are created. Combined, GDQs and DRQs all constant iteration and establish causality between possible pathways and tangible prototypes.

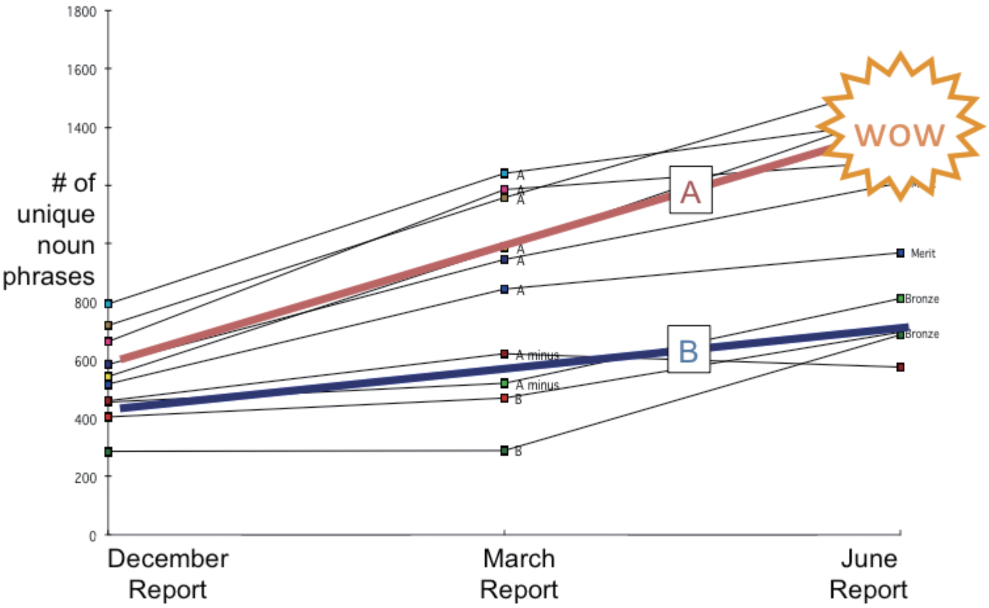


Fig. 10. After Mabogunje 1991, creative language usage, use of unique new noun-phrases correlates strongly with peer-reviewed design competition outcomes. Noun phrases describing this figure include: line-width, figure height, font-shape, legend-color, axis-label, etc.

4.3. Knowledge transmission, the power of additional channels

Product development and design projects are team based. In addition there is a close and continuous interaction with other stakeholders such as users, suppliers, etc. The success of these interactions, in our specific case the rate of learning which we want to increase, depends on how well we are able to communicate, to transmit knowledge. Besides the already mentioned impact of language (noun phrases and questions) we have (since the early 1990's) and still are studying the impact and facilitation of collaborative interactions (Tang, 1989) (Tang, 1991) (Ju, 2008). Focused onto analyzing the designer interaction through video analysis for shared workspace process activities: listening, drawing and gesturing. Additionally Tang separated between the following activity purposes: storing information, expressing ideas, and mediating interactions.

As a result, we can show that purposes and process activities intermix fluently.

For storing explicit knowledge, text is the medium of choice, whereas drawing is especially important when expressing ideas (a picture in fact says more than a thousand words?) Additionally, the eminent status of gesturing becomes obvious for conveying and supporting communication. Any workspace that is to foster change must support all three activities, allow common access for all participants and it must convey a sense of close proximity. This holds true not only for our tangible workspaces such as shop equipment but also for our virtual project platforms and computer supported cooperative work tools. It has been shown again and again that barriers posed by awkward user interfaces inhibit the exchange of information. We need to choose not the most sophisticated platform but the one actually accepted, adopted, and used the most. Especially in the age of rapidly evolving cloud services and apps, the tools most in demand may be outdated quite fast. Therefore the call for flexibility and low investments made during the discussion on physical space earlier, applies.

Based on this research we are currently exploring the possibility to enhance computer-supported communication by introducing additional information transmission, for example gesturing. David Sirkin

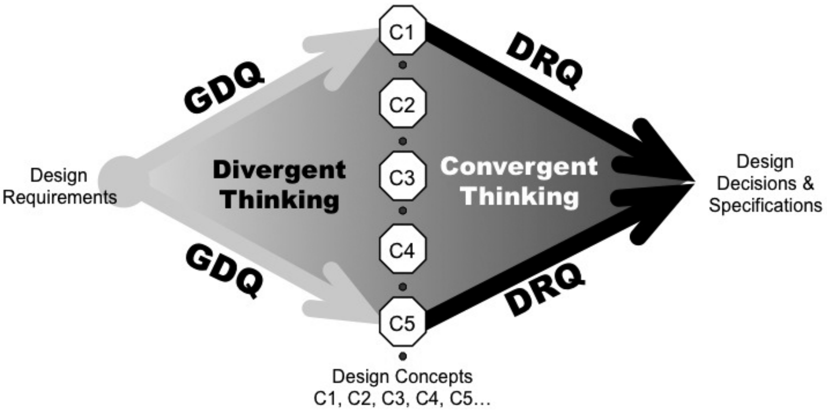
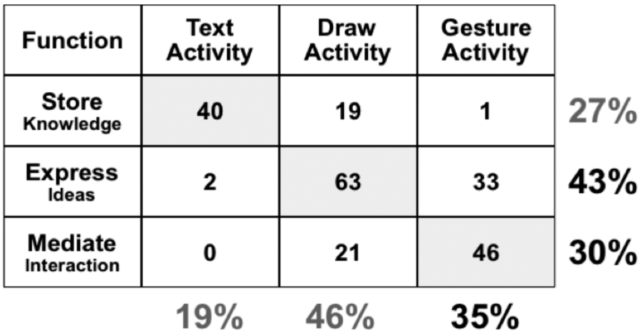


Fig. 11. (Eris, 2003), One might define a cycle of design thinking as a period of generative divergence, how many ways can we do this? Followed by an analytic convergent, given what we know, what is the next step. Real life is accordingly made up of thousands of loosely associated, one might even say fractal, elements that in agregate become space craft, groceries, banking enterprise software.



Function	Text Activity	Draw Activity	Gesture Activity	
Store Knowledge	40	19	1	27%
Express Ideas	2	63	33	43%
Mediate Interaction	0	21	46	30%
	19%	46%	35%	

Fig. 12. Tang (1989) analyzed the medium (text, drawing, gesture) used to mediate the function (store knowledge, express ideas, mediate interaction between designers) during a software development project dealing with human-computer -interaction design. Rows are functions. Columns are media. The diagonal features the strongest function-to-medium associations. The most notable finding was that gesture plays a very important role in mediation and is one of the most difficult media to capture, store, index or re-use design knowledge. Numbers on the perimeter reflect the net percentage of dialog transactions that took place in the medium or function.

has shown that the imitation of body language on behalf of the computer terminal in a human computer interaction scenario significantly improves collaboration. (Sirkin, Sonalkar, Jung, & L. J. Leifer, 2009) (Sirkin, Ju, 2011)

Support in such a way, the collaboration, though distant and computer-mediated, becomes more natural and more productive. In 2009 a team composed of Stanford and Swiss students were able to also increase the level of subjective proximity by introducing 3D audio. Sound seems to allow us to sense the spatial presence of our faraway counterparts more naturally and hence lets us focus more easily at the task at hand. Creating proximity can facilitate change and its prerequisite, collaboration. Wherever this is geographically not possible, the deployed support systems should aim to technically enhance and simulate all channels off knowledge transmission. Besides audio and visual this also includes gesturing.

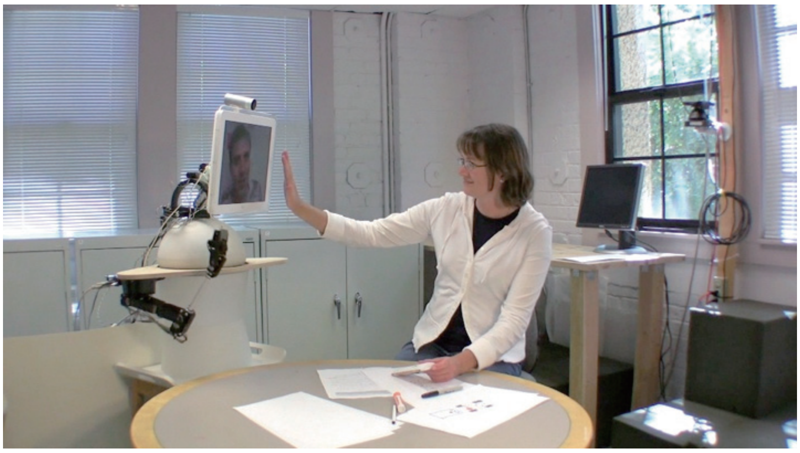


Fig. 13. The video prototypes of Study 2 depicted a design collaboration scenario. In half of the study conditions, the video showed on-screen and in-space proxy gestures (shown here); in the other half, the video showed onscreen proxy gestures only. Subjects strongly preferred the synchronous gesture/video. Not easily seen in this photo, the dark rectangle below the lower left corner of the screen is the “hand” of a very simple robot arm.

4.4. Team interaction and especially intra team conflict is emotional

The last lesson learned we would like to introduce at this point concerns emotion. It is a dimension that ought to be included into collaboration and change projects. Malte Jung has focused onto the topic of team conflict for the last three years at CDR (Jung, Chong, & L. Leifer, 2010). The initial inspiration stems from Gottmann and Levenson’s work that predicted long-term outcome of marriage. Their predictions are based on the affective interaction quality during a 15-minute video sample of a couple engaging in a problem discussion with an impressive accuracy of 93% (Gottman & Levenson, 2000). A similar problem presents itself when looking at the functionality or dis-functionality of a development team. A positive self-sustaining cycle of iterative prototype based learning assumes a working team interaction.

Using video observation and the Specific Affect Coding System (SPAFF) (Coan & Gottman, 2007) coding scheme, in this special case of pair programmers confronted with a programming design challenge, we were able to quantitatively code the affective interaction dynamics. Based on analyzing positive vs. negative facial expressions, body gestures, semantic context, and tone of voice events over time and putting them into relation with team satisfaction and performance indicators, it becomes obvious that negative events do in some teams overwhelm positive events. This may escalate up to a pivoting point where the entire team interaction becomes irreversibly negatively loaded. At this stage it is hardly possible to innovate or collaborate creatively. In fact this pivoting creates internal friction and inertia that inhibit any kind of positive learning loop. Similar to Gottmann, based on Jung’s work, we were able predict such outcomes through video coding and analysis of 15 minutes of team interaction between the pair programmers.

If team dynamics develop into this negative direction and pivot, it is necessary to intervene or to change the organizational setup. Positive change cannot be ordered from above. Emotions between team members must be taken into considerations.

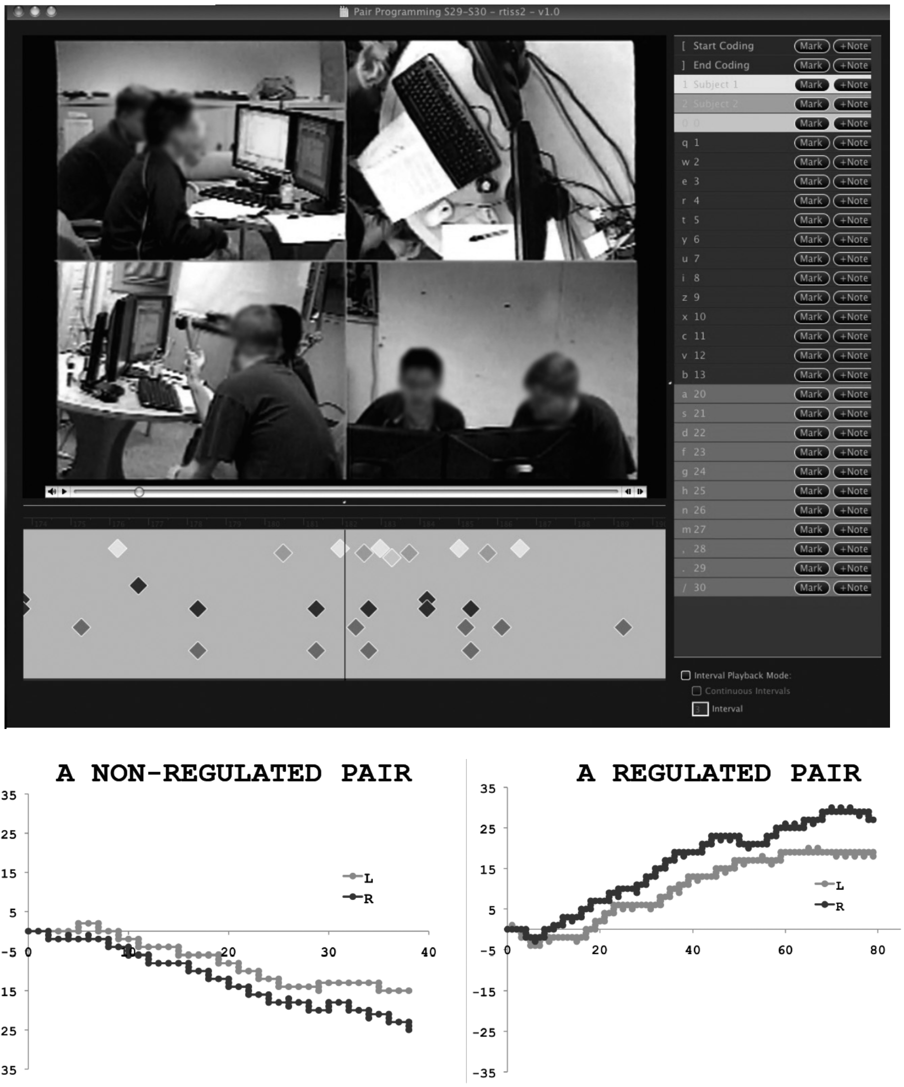


Fig. 14. Top: VCode interface showing a coded 16-second section. The orange and yellow squares in the upper rows indicate speaker turns. The red and green squares in the lower part mark occurrences of negative and positive behaviors respectively. Bottom: Example point graphs of a regulated and a nonregulated programming pair. The graphs always show the emotion trajectories for each programmer separately. The left graph is drawn from a pair that scored amongst the lowest in the sample and the right graph is drawn from a pair that scored amongst the highest of the pairs studied.

5. Dancing with ambiguity – summary and discussion

With this paper we have attempted to summarize and translate the lessons we have learned from studying designers and their activity into the broader context of complex social technical systems in general. Of course, as our research was never intended specifically for probing into this question, external validity remains to be discussed and seen. However, the aim was to show analogies and to open up opportunities for learning, to open up another perspective on the causality of change.

After translating the problem into design specific language, we have introduced learning loops that are instrumental in our understanding of change induced by product development and design teams.

On the level of formal and institutional learning, this concerns the organizational setup and procedural and managerial support systems, we advise to introduce a maximum of flexibility, and to reduce inertia by removing barriers to change such as established policies and protocol. Change as we see it implies an attempting to create something new, perhaps radically new. This requires freedom, space, organizational and institutional as well as procedural freedom and the support of the organization to actually break through the boundaries of the established knowledge.

On the level of procedural learning, we would like to emphasize that we do not have a fixed recipe-like process we follow. Instead we have identified certain activities that help new concept and new product creation, activities that substantially increase the probability of success. One is the notion of accepting rapid iterations and rough prototyping to accelerate learning and to open up the solution space. In fact we propose that there is an inverse relationship between the refinedness of the prototype material and the size of the solution space opened up by the created conceptual alternatives. Also we strongly encourage focusing on and including the user at all times and in all phases. Need finding, bodystorming, and user experience enactment are techniques that have contributed positively to the final product.

The last learning loop presented some underlying design principles and observations that might be helpful for facilitating change. The first concerns new noun-phrases and questions. The creating of new language and the usage of questions correlates with the performance of design teams. We therefore suspect that facilitating the same should produce better products and services, and maybe facilitate change itself. Questions can be separated into GDQs and DRQs, representing divergent generative thinking and convergent analytical thinking. Both are necessary but especially without the first, the level of novelty diminishes. The last two insights concern the transmission of knowledge as a foundation for change and the identification of firstly negative emotions as a barrier and secondly positive emotions as an opening to change. The proximity between the people is key for change as it allows transmitting and thus learning knowledge through a multitude of channels. We propose to physically create this proximity, and, wherever not possible, to attempt to increase the subjective level of proximity by introducing technologies such as tangible gesturing in CSCW tools. We have also learned that change activities can come to a sudden halt when the emotional interaction between players pivots into the negative. We would therefore advise to scan for such events and to counter stalemates early.

Staying within our dogma of constant redesign, and knowing that we attempt to create the unknown, we do not claim that our findings are laws in the scientific method sense, but we are experiencing their success daily. We are working on improving the techniques employed and strive to better understand the underlying principles.

We have primarily learned that in order to facilitate change, we have to let change happen. We have to remove institutional and procedural barriers, create a maximum of flexibility and support divergent activities. We have had to learn not only to live with change, but also to promote it – In this sense we would like to encourage the reader and invite him or her to join our dancing with ambiguity.

References

Alexander, C. (1964). *Notes on the Synthesis of Form.* Harvard Univ Pr.

Aquino, L. M., Steinert, M. and Leifer, L. (2011). Designing to Maximize Value for Multiple Stakeholders: evidence of the challenge to the medical-technology innovation. *Proceedings of the International Conference on Design Engineering.* Presented at the ICED 2011, Kopenhagen.

Argyris, C. and Schon, D. A. (1978). Organizational leaming: a theory of action perspective. *Reading.*

Banathy, B. H. (1996). *Designing social systems in a changing world.* Plenum Publishing Corporation.

Carleton, T. L. (2011). *The Value of Vision In Radical Technological Innovation*. Stanford University, Stanford.

Carleton, T. and Leifer, L. (2009). Stanford's ME310 Course as an Evolution of Engineering Design. *Proceedings of the 19th CIRP Design Conference–Competitive Design*. Cranfield University Press.

Coan, J. A. and Gottman, J. M. (2007). The Specific Affect Coding System (SPAFF). *Handbook of emotion elicitation and assessment*, 267-285.

Cross, N. (2000). *Engineering design methods: strategies for product design* (Vol. 58). Wiley Chichester.

DARPA. (2010, January 28). Defense Advanced Research Projects Agency. *DARPA Mission*. Retrieved from http://www.darpa.mil/mission.html.

Edelman, J. A., Banerjee, B., Jung, M., Sonalkar, N. and Lande, M. (2009). Hidden in Plain Sight: affordances of shared models in team-based design. *Proceedings of the International Conference on Engineering Design*. Presented at the ICED 2009, Stanford.

Edelman, J., & Currano, R. (2011). Re-representation: Affordances of Shared Models in Team-Based Design. *Design Thinking*, 61-79.

Eris, O. (2003). Asking generative design questions: a fundamental cognitive mechanism in design thinking. *Proceedings of the International Conference on Engineering Design* (pp. 19-21).

Eris, O. (2004). *Effective inquiry for innovative engineering design*. Springer Netherlands.

Eris, O. and Leifer, L. (2003). Faciliating Product Development Knowledge Acquisition: Interaction between the Expert and the Team. *International Journal of Engineering Education, 19*(1), 142-152.

Geim, A. K. and Novoselov, K. S. (2007). The rise of graphene. *Nature materials, 6*(3), 183-191.

Gottman, J. M. and Levenson, R. W. (2000). The Timing of Divorce: Predicting When a Couple Will Divorce Over a 14 Year Period. *Journal of Marriage and Family, 62*(3), 737-745.

Hauser, J. R. and Clausing, D. (1988). The house of quality. *Harvard business review, 66*(3), 63-73.

Ju, W. G. (2008). *The design of implicit interactions*. Stanford University.

Jung, M., Chong, J. and Leifer, L. (2010). Pair Programming Performance: An emotional dynamicspoint of view from marital pair counseling. *Electronic Colloquium on Computational Complexity ECDTR*, (2).

Mabogunje, A. (1997). *Measuring Conceptual Design Performance in Mechanical Engineering: A Question Based Approach*. Doctoral Dissertation, Stanford, California: Stanford University, 1997.

MacCormack, A. (2001). How internet companies build software. *MIT Sloan Management Review, 42*(2), 75-84.

ME310. (2011, January 30). ME310. Retrieved from http://www.stanford.edu/group/me310/me310_2010/about.html.

ME310 EXPE 2011 (class report). (2011).

Nonaka, I. and Takeuchi, H. (1995). The knowledge-creating company. *New York, 1*, 995.

PRL. (2011, January 30). Product Realization Laboratory. Retrieved from http://www.stanford.edu/group/prl/prl_site/.

Pugh, S. (1996). *Concept Selection–A Method that Works, Chapter 14 from "Creating Innovative Products using Total Design"*, Addison-Wesley.

Sirkin, D. M., Sonalkar, N., Jung, M., & Leifer, L. J. (2009). Lowering Barriers to Distributed Design Research Collaboration. *Proceedings of the 17th International Conference on Engineering Design* (*ICED'09*), *Vol. 9* (pp. 279-286).

Tang, J. C. (1989). *Toward an understanding of the use of shared workspaces by design teams*. Stanford University.

Tang, J. C. (1991). Findings from observational studies of collaborative work. *International Journal of Man-machine studies, 34*(2), 143-160.

Tang, J. C. and Leifer, L. J. (1991). An observational methodology for studying group design activity. *Research in engineering design, 2*(4), 209-219.

Thomke, S. and Fujimoto, T. (2000). The effect of front-loading problem-solving on product development performance. *Journal of Product Innovation Management*, 17(2), 128-142.

Ulrich, K. T. and Eppinger, S. D. (2008). *Product design and development*. McGraw-Hill Higher Education. Retrieved from http://books.google.com/books?id=WTNoSgAACAAJ.

Urban Dictionary. (2011). Urban Dictionary – DARPA hard. Retrieved from http://www.urbandictionary.com/define.php?term=DARPA%20hard.

Van Atta, R. H., Cook, A., Gutmanis, I., Lippitz, M. J., Lupo, J. and VA, I. F. D. A. A. (2003). Transformation and Transition: DARPA's Role in Fostering an Emerging Revolution in Military Affairs. Volume 2-Detailed Assessments.

Authors' Bios

Larry Leifer, Ph.D., is a Professor of Mechanical Engineering Design and founding Director of the Center for Design Research (CDR) at Stanford University. A member of the faculty since 1976, he teaches the industry sponsored master's course ME310, "Global Project-Based Engineering Design, Innovation, and Development;" a thesis seminar, "Design Theory and Methodology Forum;" and a freshman seminar "Designing the Human Experience." Research themes include: 1) creating collaborative engineering design environments for distributed product innovation teams; 2) instrumentation of that environment for design knowledge capture, indexing, reuse, and performance assessment; and 3), design-for-sustainable-wellbeing.

Martin Steinert, Ph.D., Acting Assistant Professor ME, is Deputy Director of the Center for Design Research (CDR) and of the Hasso Plattner Design Thinking Research Program (HPDTRP) at Stanford University. His research focuses on optimizing the intersection between engineering, new product development and the design process. Dr. Steinert teaches a seminar on "Design Theory and Methodology" and a project based course on "Designing Emotion Interactive Interfaces". Three of his four current sponsored research projects combine engineering design behavior and real-time physiological measurements.