

# New Telco

*Practices for transforming traditional Telcos to an Internet Protocol world*

version 0.3.1



Steam-bent wood construction by Paul Timmer



## Legal Notice

### Copyright:

The copyright for this document is the property of Product Foundry. This document is supplied on the expressed condition that the content must not be used for purposes other than that for which it has been supplied, or reproduced, wholly or in part without the prior written permission of Product Foundry.

### Disclaimer:

The views expressed are purely those of the authors and may not in any circumstances be regarded as stating an official position of Product Foundry. More information on the specific activity treated in this publication is available on the Internet, and is supplied with the conditions outlined before downloading a copy of this document.

<http://www.patterndeck.com/>

### Contact Information:

info@product-foundry.com  
Product Foundry, Haarlemerweg 10,  
1014 BE Amsterdam, The Netherlands

### Applying the Pattern Language:

The contents of this document are intended for application by experienced practitioners within Telcos, and the content provided by this document such should be sufficiently self-explanatory. However, all applications, derivative action, and decisions arising from the contents of this document are entirely at the risk and discretion of the practitioners. If you require assistance, we recommend engaging Embodied Making certified analysts and certified New Telco Pattern Language practitioners.

## Recent Changes

### 0.3.1

---

-  added placeholders for [Piecemeal Provisioning](#), [Real Time Offers](#), [Rhythmic Sensemaking Spaces](#), [Usage View as it Occurs](#), and [Visible Assists](#)
-  changed [Decide by Matching](#) , [Own Base Currency](#)

# Table of Contents

## Patterns by Name

Adapting with Life	Aware Instances	Change History	Changeable by Many	Classic Lines	Clear Withering Products	Configurability inverse to Complexity
Constant Circulation	Continuous Harvesting	Customers create their Own	Decide by Matching	Declarative Component Dependencies	Deep Roots	Everyone Cares
Events Everywhere	Everyone's Business	Exotic Plants	Few Skilled Gardeners	Fortified Town	Growth Supports	Infinite Varietals
Interdisciplinary Expertise	Integral Technology Stack	Language for Product Instances	Legacy Agents	Living Data Everywhere	Living Prototypes	Low-walled Town Communities
Many Products Co-exist	Meaningful Names and Identities	Multi-criteria Directories	Nurture Keystones	One just like that One	Own Base Currency	Participating Demographic
Piecemeal Provisioning	Plant in a Single Garden	Pre-provisioned Frequents	Product Markers	Productpedia	Provider Porting	Pruning Products
Real Time Offers	Recast Molds	Repurpose Infrastructure	Ringed Growth	Rhythmic Sensemaking Spaces	Several Prototypical Configurations	Shade from Others
Signature Aromas	Specify to Operationalize	Temporary Splints	Themed Areas	Three Product Champions	Town Well	Trusted Circles
Usage View as it Occurs	Visible Assists	Whole Town Participates				

## INTRODUCTION

Incorporating changes in our individual lifestyle is difficult, whether it be learning new skills or changing our habits. Incorporating meaningful change in large enterprises with several individuals is exponentially more difficult. It isn't a single solution applied by a single individual that changes a large and complex enterprise, but several solutions applied simultaneously and coherently by several individuals. Given that products unify an enterprise — in that individuals collectively design, produce, sell, support, operate, and decommission them — we offer a set of solutions to transform enterprises on the basis of products. Pattern languages, which were originally conceived by Christopher Alexander as a means of describing effective practices within architecture, can be used to represent the knowledge required for product lifecycle management across a large enterprise. We do this by understanding the prevalent forces in the enterprise and finding positive solutions or patterns that are able to balance these forces.

## TELECOM COMPLEXITY

Telecom operators are confronting business challenges that require deep changes within their people, systems, and organizations. With device manufacturers such as Apple, Google, and Samsung increasingly providing their own communications services, telecom operators are facing declining revenues from their staple offerings of voice, data, and messaging. The emergence of highspeed mobile Internet coupled with devices that offer users greater possibilities in service selection and consumption requires telecom operators to build on their existing strengths while capitalizing on new opportunities. Products focused around voice, data, and messaging need to be transformed and merged with new services that revolve around the connected work and life of users. The omnipresence of Internet Protocol (IP)-based networks makes it easier to create high volumes of diverse multidevice products, which will increase the complexity in the conception, implementation, operation, and retirement of products. Current inefficiencies will be further amplified given the product volume and diversity, and new practices will need to emerge in order to manage this unprecedented scale of complexity. Each of these new practices will have to be applied with other practices, creating an interconnected network of practices. These practices can help telecom operators effectively transform from value chains to value networks. Telecom operators have traditionally tried to manage product lifecycle processes through individual and independent projects, using waterfall or agile management methods. Products realized in environments that use a waterfall methodology have distinct phases for product innovation, product feasibility assessment, product implementation, product operation, and product retirement. Matrix organizations are often constructed to complement these distinct phases, and particular responsibilities are assigned to each phase. Distinct organizational units within the matrix organization have clearly defined roles and responsibilities, each producing a specific aspect of the final product. A single department that translates all business needs into designs (or a single group that manages all the data in the enterprise) is an example of these distinct organizational units. Such structures are useful for producing products that have similar characteristics.

These market trends were predicted as early as 2006. See Peppard, Joe, and Anna Rylander. "From Value Chain to Value Network: Insights for Mobile Operators." *European Management Journal*, Vol. 24, No. 2, 2006.

Historically, most commercial product offerings have had similar structures, making it possible to create relatively repeatable processes for their lifecycles. The focus of such organizational structures tends to be around product delivery rather than product retirement. These processes are analogous to the classic Ford Model T assembly line, in which each person in the assembly line had a defined role, received fixed inputs, and produced fixed outputs to achieve consistency. Figure 1 illustrates the treatment of product lifecycle processes in a waterfall method.





Figure 1 — Phases in the lifecycle of a product within a waterfall methodology.

Products realized in agile environments follow more fluid and flexible phases, where each phase has a tendency to overlap with other phases. There is recognition that features that seem very important for a product at an early stage might seem less important at a subsequent stage. Through understanding the complexity of tasks to realize a product, development can be prioritized by tackling the most complex aspects of a product first. Documentation and service-level agreements are kept down to a minimum and replaced by incorporating suppliers and partners as extended members of the production system. Processes themselves are constantly being adapted in the face of changing situations by practitioners who continuously apply improvements in their methods of working, following principles like “5 whys is 1 how.” However, agile work environments are difficult to achieve in highly distributed teams within complex organizations. The inspiration for agile methodologies can be traced to the Toyota Production System as realized in the late 1970s, which introduced the Kaizen philosophy of production. Figure 2 illustrates the treatment of product lifecycle processes using agile methods.

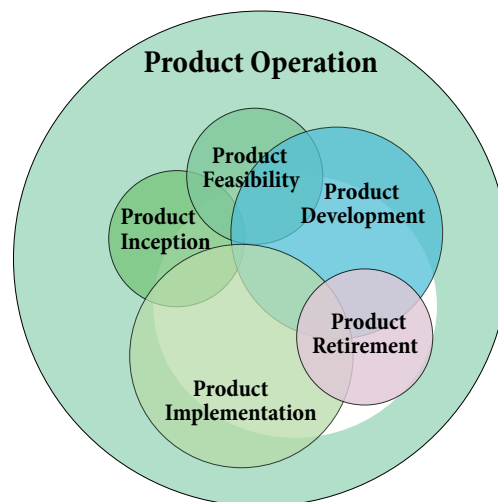


Figure 2 — Phases in the lifecycle of a product within an agile methodology.

Both waterfall and agile methods provide practices for structuring organizations for efficient product lifecycle management. There are other practices that can be applied to managing people, technological choices, organizational structures, organizational culture, and physical workspaces. These practices can be selectively applied in different combinations based on different situations and organizational cultures. The practices themselves can be classified based on their maturity. Best practices are typically established in situations where the correlation between cause and effect is obvious to all. Good practices require some analysis to establish a stable relationship between cause and effect. Finally, emergent practices typically come about when retrospective analysis is done on the correlation between cause and effect.

A pattern language can be used to describe such practices, indicating best practices with three stars and good practices with two stars. (If we had featured any emerging practices, we would have assigned these one star.) We first give a synopsis of the scale of complexity telecom operators confront today in each phase of a product’s lifecycle. Next, we provide two examples from our pattern language, outlining the problem the pattern solves, the pattern’s solution, and the linkage of this individual pattern with all other patterns. By using a pattern language, these practices can be woven in myriad ways for different products within the same organization, and the unique challenges a particular telecom



operator faces can be resolved using a unique implementation of these patterns.

## DERIVING THE PATTERN LANGUAGE

The primary goal of developing a pattern language for telecom operators is to provide a flexible method for better dealing with the scale of complexity they confront in using new practices. The adoption of these new practices can require new organizational structures, changes in employee roles, the creation of new employee roles, and the gradual removal of existing practices. All these requirements for the adoption of new practices can be very difficult to achieve if the benefits of the change are not clear at every level of an organization. Patterns of change must be able to incorporate existing practices by absorbing their strengths and minimizing their weaknesses. Each pattern must be able to articulate the existing problems based on real situations individuals encounter, the solution for these problems, and the linkages of the pattern with other patterns.

In order to derive this pattern language, we have applied the [Embodied Making](#) method. The process of design with embodied making is initiated by having conversations with people in the space where we want our designs to live. It starts by capturing stories, which are anecdotes as people relate them to us, and then faithfully recording them. We conducted around 120 interview sessions with people performing different roles in these organizations, ranging from customer service agents to product developers. All these individuals were involved in some capacity in the realization, operation, or retirement of products. From these interviews, we were then able to derive the prevalent forces from the processes of product lifecycle management. For instance, the following are some of the forces evident during the inception of a product:

- Reluctance to discuss immature ideas
- Fear that ideas will be underappreciated
- Concern that competitors may copy ideas
- Desire to obtain idea feedback from experts
- Inability to identify experts
- Tendency of experts to prefer giving direct and personal feedback
- Tendency to forget about ideas after a while
- Skepticism of unconventional or disruptive ideas
- Tendency of commercially successful products to get more attention

We have thus far identified around 800 individual forces that are common in most Telecommunications Operators. These forces did not manifest themselves in isolation but rather in combination with many of the other forces. For instance, one reason for the reluctance to discuss immature ideas is the fear that the ideas will be underappreciated. A solution for resolving these forces would be to create an environment where anyone can submit an idea anonymously. However, this solution conflicts with other forces, such as the desire to get credit for good ideas and the fear that others will get credit for ideas that are not their own.

Another solution that resolves these forces is to create a forum where individuals can choose to submit their ideas either openly or anonymously, with the option to reveal their identity at a later stage. However, anonymity doesn't allow individuals to receive direct and personal feedback from experts, and it doesn't resolve another force — the tendency for a single individual to have limited insight about all aspects of a product. A solution that does resolve these additional forces is to require that for any idea to be adopted, it must be championed by three individuals who collectively consider its commercial, operational, and technological aspects. This continuous process of assessing solutions eventually provides a set of stable solutions that work well together, and each of these solutions is then restructured into a pattern. Wherever possible, the patterns are named after a common theme, such as a garden or town. As new patterns emerge, existing patterns are reassessed and redesigned to reflect their individual and collective interaction with other patterns. Together the patterns provide an interwoven pattern language.

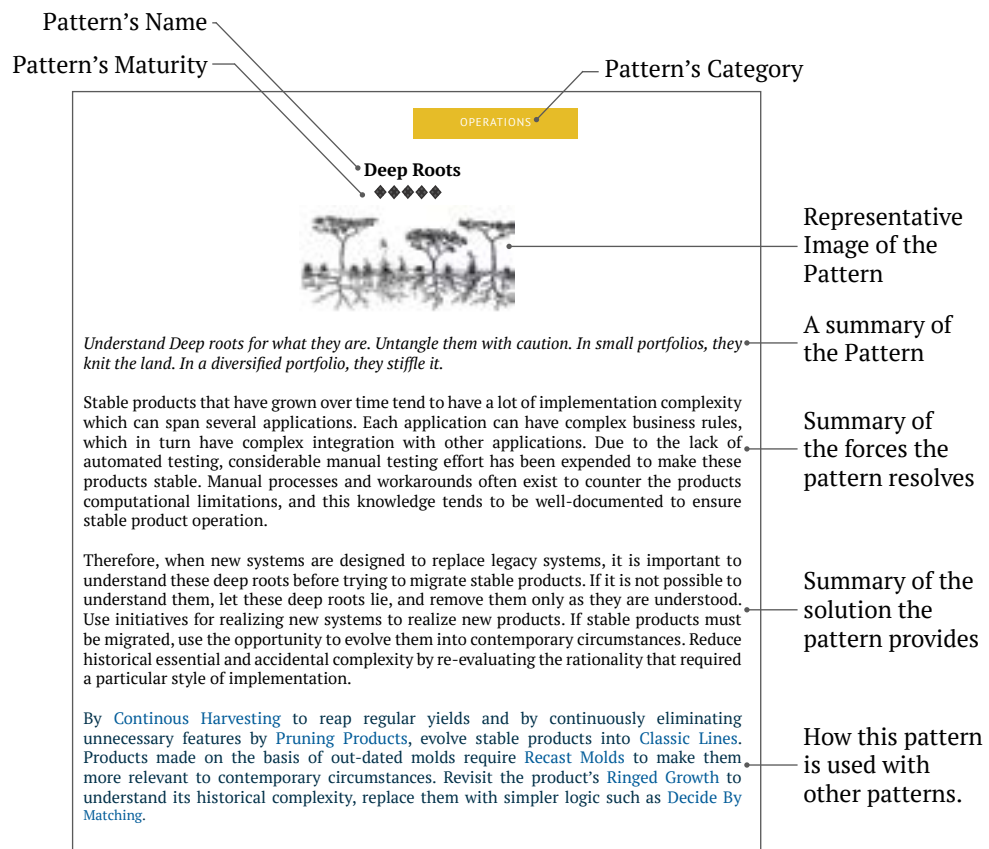


## HOW TO READ PATTERNS

There is no sequence to reading the Pattern Language, and it is possible to arbitrarily select any Pattern within this collection and makes sense of it in terms of the problems and challenges proposes to solve (forces), the nature of the solution (pattern), and it's relationship with all other patterns. Each pattern starts with a categorization, and for the moment, all patterns are either categorized by Organization Structure (■), Market Strategy (■), Operations (■), or Component Behaviour (■). This categorization reflects the salient purpose of the pattern, and it doesn't mean that a pattern about Component Behaviour does not have some impact on Organization Structure. Patterns about Organization Structure are principally about how departments and leaders should organize work within a Telco. Market Strategy patterns are principally concerned with customer and brand experience, and the nature of propositions to customers. Operations patterns are principally concerned with the internal practices of anyone within a Telco. Finally, Component Behavior defines how technology should be configured and utilized. This doesn't mean that only technical people need to understand Component Behaviour, or that only marketers need to understand the Market Strategy. By understanding these patterns, everyone in a Telco, and in some cases its Customers or Suppliers, should have an understanding how a set of patterns being applied in one space influences others.

### SITUATIONS TO APPLY THE PATTERN LANGUAGE

There are recurring situations in Telcos, such as the introduction of a new Product or the replacement of a legacy system, where the patterns.



Each pattern has a memorable name, and wherever possible, we have used names inspired by gardening or urban landscapes. The use of a common metaphor makes it easier to relate patterns in terms of each other, although we have struggled to give names using a single metaphor. Next, each pattern can have different scales of maturity, indicated by diamonds (◆). A single diamond means that the pattern is experimental and practitioners should apply the pattern through cautious probing, retrospectively understand its consequences, and then apply in some measure again. Two diamonds imply that this pattern has been successfully applied a few times, but isn't part of any existing practice. Three diamonds imply that the pattern is an existing emergent practice, and several thought leading organizations are currently attempting it. Four diamonds imply that pattern is stable practice, and has been attempted several times successfully. Five diamonds imply that the pattern is stable and should be considered the standard practice. The diamonds should not be used to understand the patterns in terms of risk, and where in the interests of



risk aversion, only patterns with five diamonds are applied. The Pattern Language is most effective when applied collectively and cohesively. The diamonds exist for practitioners to understand the measure of reflection, both in the moment and retrospectively, required from each pattern.

Next, the essence of the pattern is summarized in one or two sentences, giving those who want to skim the content or remember a previously read pattern the means to understand the pattern quickly. Next, the forces the pattern resolves are summarized. Forces influence how people experience a space, and are derived from applying Embodied Making. To quote the method:

*"Forces are things that shape their environments to be the way they are. They give birth to experiences, such as the forces that make us book the holiday of our dreams at great expense, provide twists in the flow of an experience, such as the forces that choose a product to purchase from a variety of available products, sustain certain experiences over time, such as the forces that make us form orderly queues while waiting to be served, or bring an experience to an end to see the beginning of another, such as the motivations behind walking away before a performance has finished. Forces can be reflected in motivations, such as the desire to be compensated for providing a service. They can be statements of fact, such as the presence of a high population density. They could be hopes and anxieties, such as the desire not to lose our baggage while traveling or meeting a taxi driver who charges us fairly in an unfamiliar city. They can reflect dominant values, such as the desire for security or an aversion to feeling vulnerable. Some forces are rational, tangible, or measurable. These forces follow the logical structures of cause and effect, and their effect on the environments where they exist can be reasonably predictable. For instance, the repeated failure of a service over time will see falling demand for that service, and making a service extremely reliable comes at a great cost. These are incongruent forces that can be balanced with judging the right service level for the price customers of the service are willing to pay. Other forces are often irrational, intangible, or unmeasurable, such as personal beliefs and prejudices, but are extremely influential in determining the course of experiences and nature of the systems within which they reside. For instance, if the objective of the design process is to create an object to facilitate communications, and there exists a general cultural aversion to specific colors ("black is an unlucky color") or shapes ("sharp edges are bad feng-shui"), then these are forces that will influence the form of the solution." Extracted with permission from EmbodiedMaking.org*

We have often heard it said that leadership and managers in Corporates require everything summarized in powerpoint slides with no more than three bullet points. In our experience, most individuals in these positions occasionally read a novel or newspaper, and desire explanations that address brevity and clarity in a meaningful balance.

Next, a solution that resolves and balances these forces are described. The solution may be a practice or it may be tangible entity. It is the intent of these solutions to create a positive collective impact for Telcos and their Customers. Forces may also be resolved in the favor of individuals but unfavorably for the collective, and these practices are known as anti-patterns. Our hope is to use these patterns to create positive working spaces within Telcos, and therefore we have avoided structuring any anti-patterns. Finally, the relationship of the pattern is established with other patterns in terms of how they should be practiced collectively, by inter-purposing in sentences.

This body of work is constantly evolving, and we are continuously adding new patterns to the language. Our view in writing these patterns is to accept their incompleteness, impermanence, and imperfection, and we publish them as they occur. We leave it to practitioners to judge whether they are in a form worthy of being practiced.

## Adapting with Life



*We cannot predict the turns our lives will take, and our Telco Products adapt to changes in our lives rather than us adapting our lives to our Telco Products.*

The majority of Telco products are structured around rigid contracts for fixed durations as subscriptions. Telcos in some countries have historically provided Customers subsidized devices. The subsidized devices are provided in exchange for a commitment to a subscription for a fixed duration with minimum monthly charges to cover the device's cost. In contemporary times, where the prices of a Smartphone is comparable to the price of a laptop, the minimal monthly charges can become higher than a Customer's projected consumption. Customers are increasingly willing to purchase their own Smartphones and only require access to calling, messaging, and data. If there are deep-rooted changes in a customer's life, such as losing their job or going through a divorce, their contracts usually remain the same. Customers are forced to project their contemporary circumstances for the duration of the contract, and purchase products based on their "best-case" consumption. Once they are into this contract, they feel locked, and are unable to make changes easily. Rather than a communications or lifestyle product that is responsive to changes in a person's life, it feels like purchasing a product from a Bank or an Insurance company. Telco products carry the impression of being loaded with heavy contractual terms, and miss opportunities to sell smaller one-off purchases, such as a single song without a music subscription, an hour's pass for high-speed internet, or a 1 week all you can call rate to an international number. Customers cope today with other solutions from other providers where they do not feel locked in, such as paying for an hour's worth of WiFi access with a hotel in a foreign country despite having a mobile subscription.

Therefore structure Telco products to exist on a base platform where subscriptions can be added or removed at will. A subscription for unlimited calls can be replaced by a subscription for 200 minutes of calls per month at half the price with a few interactions in the space of seconds. Subscriptions become recurring unless explicitly stopped, and they can be stopped at any time. If the customer has a subsidized device, give the flexibility to add and remove subscriptions at will as long as their total charged subscription rate and consumption equals their minimum required commitment. Customers are able to purchase one-off products at competitive prices, such as a single TV episode, an hour's high-speed internet access, or a week's pass to call an international number. In difficult moments of a customer's life, such as the moments when they lose their job or a loved one, give them the complete flexibility to tailor their subscriptions in any way they want, to the extent of removing all subscriptions. Provide clear communications to customers that explains the reasons for the flexibility being the desire to develop empathy with the customer's situation.

Understand the nature of change through the stories of the [Participating Demographic](#), and correlate situations to product structures in the [Productpedia](#) with [Ones just like that One](#). Understand a customer's exact situation through examining the [Aware Instances](#) and their [Change History](#), and assessing their loyalty with how much they possess in the [Own Base Currency](#). Give more freedom than usual in these situations so [Customers create their Own](#) install base, and suggest [One just like that One](#) as alternatives.

The Dunning process used by most Telcos is reminiscent of how Banks and Insurance companies deal with defaulters. See ["The worst mobile phone offenders - and how to complain."](#)

WiFi is increasingly becoming a free commodity. See ["The Price of Staying Connected."](#)

## Aware Instances



*Components are aware of their role in fulfilling products, and product instances contain their own unique semantics. Awareness isn't elsewhere, but with the component itself.*

A product instance, or a product as a customer experiences it, is realized through several systems, ranging from configuring access in a network switch to a customer record in a customer-relationship management system. Each of these systems have a part in constructing the customer's experience of the product. Network equipment in telecommunications environments historically have a lot of variation, often supporting standards from the 1980s such as GSM to contemporary standards such as LTE. With the move to an all Internet Protocol (IP) network, where there will be an even greater diversity and volume of products with providers moving beyond voice and data to content (music, movies, books, education, etc.), several configurations will be possible in each of these systems. This will lead to high-levels of complexity in managing all of these combinations. Processes for provisioning products need to anticipate several eventualities, to the extent that process definitions need to be conscious of the state of several network elements and systems simultaneously. When controllers are used to control the number of states, the controllers need equal or greater knowledge than the total number of states. Similarly when product, customer, and resource specifications are used to define the behaviour of instances, the specifications need knowledge equal or greater than the sum of instance variations. When provisioning processes are treated as end-to-end transactions, errors in the process necessitate restorations of original system configurations before the product was provisioned through an expensive rollback. Rule-bases for individual situations such as product pricing (promotions, discounts, fees, allowances, commissions, etc.) or permitted products given an individual's installed product base further increase complexity. In the highly distributed environment of telecommunications organizations, several components are able to change each other's states, and controllers and specifications need to consider meaningful combinations of state. However if the controllers and specifications are not aware of a combination of state, which can number in the several thousands, systems recovery can only be established through restarts. Given that stable systems require their control mechanism to address as many variations of state as possible, creating centralized controllers and single points of truth only increase the complexity, not reduce it. "Things fall apart, the centre cannot hold."

Therefore make product instances exist integrally where they have knowledge of the components that realize them, and conversely, the components that realize them have knowledge of the product instances that reside within them. Instances are aware when they are unfulfilled, and are capable of being fixed individually. Make it possible for instances to have individual and unique behaviour, and for customers to have their unique situations reflected in the system. When systems aren't capable of enabling unique behaviour for instances, utilize proxies for in systems with that capability.

Use the [Language for Product Instances](#) to reflect product instances and the customer's situation and use the same [Language for Product Instances](#) to interpret the customer's unique situation and install base. If the customer is experiencing problems with their product, permit customer agents to [Changeable by Many](#) within the [Fortified Town](#). Enable [Product Markers](#) that describe the product instance's performance, costs, and efficiency.

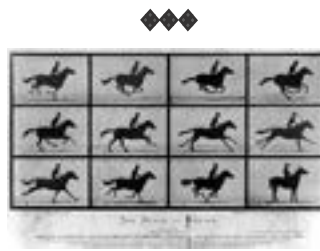
Ross Asby, the creator of the law of requisite variety, succinctly summarized "variety can destroy variety". Ashby, W.R. 1958, Requisite Variety and its implications for the control of complex systems, Cybernetica (Namur) Vol 1, No 2, 1958.

For a thorough analysis of distributed state control, see Moseley, B., Marks, P., Out of the Tar Pit, 2006.

From the opening lines of "The Second Coming" by W.B. Yeats: Turning and turning in the widening gyre, The falcon cannot hear the falconer; Things fall apart; the centre cannot hold;



## Change History



*Changes and the reasons for changes in Products, Customers, Services, and Infrastructure are always remembered, and changes can be retrospectively interpreted and understood.*

Telcos today tend to operate with standardized specifications for products, customers, services, and infrastructure (also called resources). Each of these specifications cater to a particular number of possible states. For example, a product offered as a subscription can be offered for different durations of 15 days, 1 month, 6 months, 1 year, or 2 years, resulting in 4 possible states for subscription duration. The subscription could be activated immediately, after a fixed duration (e.g. within 30 days), or on a specific date (e.g. 25 December), resulting in 3 possible states. Combinations of possible resulting states are treated in specific rules, for instance the requirement for activations of subscriptions less than 6 months to only be offered with immediate activations. For 4 possible durations, and 3 possible activations, we see 12 possible ways to consider durations and activations together. Once activated, the product could also be deactivated, where there could be 3 possible values for deactivation (immediate, after a fixed duration, or on a specific date). Once again specific rules can exist once again for deactivations, and now we have 36 possible ways to process durations, activations, and deactivations together. For example, if a product has been activated less than 10 days ago and it is for a duration of 6 months or less, it cannot be deactivated. The sequence in which rules are applied also needs to be considered. For example, it is not possible to deactivate a product if it has never been activated. With an increasing number of variables, the number of possible resulting states, how they need to be considered together, and the sequence in which they need to be considered only increases. Given the broad variety of possible states in product, service, infrastructure, and customer data, total number of possible combinations (when the sequence doesn't matter) or permutations (when sequence matters) becomes a number in the several millions. In most contemporary Telcos, when products, services, and infrastructure are changed for customers, they are changed primarily using the customer's current state, where all customers with the same current data are treated in the same way regardless of how they arrived in their current state. For example, a customer who has had a monthly subscription for 10 years with an operator (post-paid) who chooses to switch to a pay-as-you-go scheme (pre-paid) for temporary financial difficulties is henceforth treated exactly the same way as all other pre-paid customers disregarding their existing history with the operator. When they would like to switch back to being a post-paid customer, they are treated with the same criteria as a new customer. Using only their current state and using standardized specifications and rules for their current state, the potential of products, customers, services, and infrastructure is reduced to a few manageable states. Given the highly distributed nature of Telco processing systems, when the combinations or permutations of state unaddressed by those few managed states, the only option is to revert states a known stable configuration ("please restart your machine"). In addition, the storage of previous states, though technically possible, has been expensive given the high costs of processing and storage. With parallel processing architectures, high-speed storage (SSD drives), and large memory access, these historical reasons are increasingly fading away.

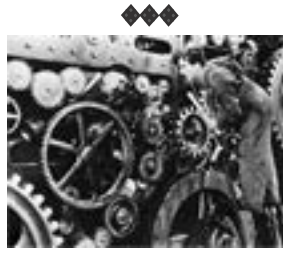
Therefore record the history of changes for products, customers, services, and infrastructure using a small set of data and knowledge structures. Using an expression language that reflects this data and knowledge structure, compose any combination that is meaningful as a whole. Let people judge which combinations make sense and don't rather than trying to transcribe this as computational logic. of a product, customer, service, or infrastructure, visibly show their history individually and in relation to other entities. Make this information easy to

understand by people and by computational systems. It is preferable to record historic state through deltas in order to reduce waste in computational storage and processing. Finally, there will always be a minority that will exploit this flexibility with malicious or harmful intent, and guard against these exploits by creating complete visibility of who and how changes were made, the reasons they were made, and correlating interpreted data and knowledge to known malicious or harmful intent. Create the space for individuals to create their own combinations, and make deviations more visibly noticeable.

Using a [Language for Instances](#), store each individual change in products, customers, services, and infrastructure with the ability to process current and previous states while considering how to treat situations, making it possible to [Adapt with Life](#). Make it possible for product, service, customer, and infrastructure to be [Changeable by Many](#). Make user experiences in systems for the [Few Skilled Gardeners](#) to monitor deviations, and make deviations prominently visible. Create open [Rhythmic Sensemaking Spaces](#) where the [Whole Town Participates](#) within the [Fortified Town](#), and secure [Rhythmic Sensemaking Spaces](#) for the [Low-walled Town Communities](#).



## Changeable by Many



*Any product stakeholder, irrespective of whether they are from marketing or technology, should be able to configure the most frequently occurring changes themselves.*

Modifying the behaviour of a product's feature, adding new features, or removing unused ones can de-stabilize the product when the dependencies between features are poorly understood. In demand-supply organizations, the problem is further amplified when changes have to be specified by the demand organization in great detail, approved by the service management organization to ensure that the operations of the existing product is not de-stabilized in any way, and finally accepted by the supply organization when the nature of the change is clearly understood. Things that seem important at the beginning of a product lifecycle seem less important after the product has been in operation for a few years, which is reflected in different change requirements for the product. The individuals who demand changes with a commercial focus usually specify what must be changed, rather than executing the change themselves. In complex environments with several products, comprehensive and detailed specification is required to make product changes, and the knowledge of how to make these changes is only understood by a few individuals in the supply organization. Only the most critical changes are implemented, and retrospective minor improvements are difficult without going through an expensive and laborious change management process. Excessive configurability introduces greater complexity, and the introduction of greater complexity without a proportional increase in value gradually chokes an organization's growth.

Therefore avoid making the product excessively configurable when it is first being realized, and focus on ensuring the product's dependencies are clearly understood. Once the product is in operation, identify the few changes that are required most of the time, and make those configurable. Make configuration knowledge necessary for every product stakeholder, regardless of whether they have a commercial, implementation, or operational focus. Focus on the configurability of an individual product first, rather than seeking structures to configure several products simultaneously. Delay "future-proofing" the product's configurability by trying to predict the aspects of the product that might require configuration someday by focusing on existing patterns of configurability. Introduce configurability in a manner that commercial, technical, and operational staff can make changes. If a product really requires a high quotient of configurability, prefer making changes in source code over configuration files.

Make a product's **Configurability Inverse to its Complexity**, and ensure investment in a product's configurability develops through its **Ringed Growth**. Make the salient features of the product configurable by any of the **Three Champions**. Through **Constant Circulation** where all products are **Planted in a Single Garden**, and continuously **Pruning Products** by a **Few Skilled Gardeners**, keep the configurability of products proportional to their value.



## Classic Lines



*Recognize, foster, and grow Classic lines. Keep a few of these moulds. Some products never grow old. Not everyone wants the latest toy. It just looks that way to technocrats.*

The Telecommunications industry is characterized by frequent technological advancements which enable new products. These products enable faster speeds, improves the quality of communications, and permits the convergence of telecommunications services utilizing common enabling technology. While technology-savvy customers are eager to try out emergent technology, there exists a customer base for more traditional products. For instance, customers who prefer mobile devices that just make phone calls but have long battery lives are increasingly faced with fewer and fewer choices. In some cases, the need to migrate old technologies and consolidate it with newer technology leads to initial levels of poor service quality and unfamiliarity with the product, such as the case of migrating customers using voice with ISDN access to voice with IP access. Withdrawing older products with loyal customers can lead to losing them despite giving them newer alternatives based on advanced technology.

Therefore identify products that have loyal customer bases, display stable patterns of consumption and usage, and provide user experiences that distinguish them from other products. Create virtual and physical spaces where users can share their passions about products with other users. Understand the situations where customers use these products, and their utility as communications enablers. When changing the underlying enabling technology to another, try and recreate the user experiences in forms that find a balance between the new technology and existing ways of work. Avoid excessive skeumorphic design in trying to preserve classics, such as preserving rotary dials on digital telephones [1], but also avoid radical new forms that depart from existing forms without providing sufficient utility. If a classic product cannot be continued, migrate classic users by explaining how alternatives preserve the key characteristics of these classics.

Understand a product's usage patterns with the [Participating Demographic](#), and encourage the [Three Product Champions](#) to determine timeless characteristics of their products as [Signature Aromas](#). Expand on this understanding by speaking to users in the [Low-walled Town Communities](#) for product initiatives where the [Whole Town Participates](#). Create internal product fan clubs around the [Town Well](#), and owner fan clubs beyond the walls of the [Fortified Town](#).



## Clear Withering Products



*When the product starts to wither, clear it and replace it with new products.*

Telecommunications products are launched within a specific market context which can consist of specifically identified consumer needs and wants, prototypical consumer behavior, and required consumer education and security. If the gap between the product's original purpose and the contemporary circumstances is far too broad, the product's market penetration and sustainability will start to wither. Attempts to regenerate these products with broad gaps require revisions in their original marketing and operational strategy. These products with broad gaps also tend to have small and shrinking customer bases, and their continued operation incurs low or negative profitability. Despite best intentions, products that are neglected due to shifts in organizational focus tend to wither over time. Although these products are soon removed from commercial catalogs to avoid new customers, they continue to be operated for existing customers leading. This leads to reduced organizational focus, brand dilution, reduced overall profitability, and increased operational complexity.

If a product's market reach and customer base shrinks to the extent that it incurs low or negative profitability, first explore the possibility of whether it can be easily transformed or altered to improve its commercial sustainability. If the transformation of the product requires greater effort than its market potential, remove the product. If the complete removal of the product is extremely difficult due to its implementation, create operational procedures to demarcate its non-operation, and deprecate these practices when migrating to new systems.

Use [Product Markers](#) to understand the criteria for a product's success, and through [Constant Circulation by a Few Skilled Gardeners](#) where all products are [Planted in a Single Garden](#). When [Product Markers](#) indicate a product's decline, engage the [Three Product Champions](#) and the [Participating Demographic](#) to understand the original market context and the reasons for the product's decline. Attempt to transform the product based on [Recast Moulds](#) and utilize [Growth Supports](#) if the product exhibits potential. If all else fails, remove the product from all operational and commercial activities. If the product has [Deep Roots](#), let it lie and deprecate it gradually from commercial and operational activities.

## Configuration inverse to its Complexity



*If a product has a few configurable options, configure using a proprietary language. If a product is highly configurable, configure using source code.*

Products rarely have a stable set of characteristics, and their characteristics and behavior need to be modified continuously throughout their operation. These changes can range from price alterations, taxation adjustments, device dependencies (e.g. change the device offered with the product), promotional offerings, to modifications in enabling technology. The ability to modify a product's characteristics and behavior during its operation is defined as its configurability. Each configurable option introduces an alternate variable state, and each possible state introduces greater complexity. Business stakeholders often desire their product's to be easily modifiable, increasing the product's configurability requirements. The introduction of each configurable option increases the system's complexity and maintenance becomes more difficult and expensive. Systems with several configurable options require specialized knowledge, which tends to become entrenched with a few individuals. Similarly, vendors offering proprietary systems that are highly configurable require specialized knowledge which is more difficult to obtain than those of open languages.

For a thorough analysis of distributed state control, see Moseley, B., Marks, P., Out of the Tar Pit, 2006.

Therefore introduce product configurability extremely cautiously. If a single system requires a few configurable options, offer configuration through proprietary configuration settings. If a single system requires several configurable options, manage their configurations directly through source code developed in open languages. Avoid the use of central controllers that configure several systems simultaneously, and build awareness of configurations spanning multiple systems within each of those systems individually.

The [Few Skilled Gardeners](#) and [Three Product Champions](#) collectively decide what to make configurable in a system, avoiding excessive configuration whenever possible. The configurability of a product is altered throughout its [Ringed Growth](#).

## Constant Circulation



*Always have a few new products being introduced as a few products are being retired.*

Given the virtual nature of telecommunications products, it becomes possible to easily release several variations of the same product with different pricing structures for customers to create product diversity. Consequently, most telecommunications operators tend to have large volume of products, regardless of whether products are counted on the basis of distinct types of commercial offerings or on the basis of underlying enabling technology. The boundaries and distinctions between products are often difficult to comprehend, whether they exist in commercial catalogs, information systems such as customer-relationship management systems, or in network elements like switches. The introduction of new products becomes difficult due to the capacity constraints from operating existing products.

Therefore create constant circulation where as a few products are being introduced, others are being removed. Maintain the configuration and implementation of individual products as distinctly as possible in order to reduce complexity in their removal. Avoid the merger of many products into a single product line where the distinctions between products are lost, and maintain distinctions between high performing products and poorly performing products.

Have a [Few Skilled Gardeners](#) tend to products [Planted in a Single Garden](#), while constantly [Clearing Withering Products](#) and introducing new products with [Growth Supports](#) that receive [Shade From Others](#).

## Continuous Harvesting



*A product, once planted, is harvested continuously. In projects, it would be forgotten once planted.*

When products are first implemented and launched, they receive sales, marketing, technology, and operational focus. Sales channels and marketing campaigns are focused on making the new product commercially successful and embedding it in the consciousness of potential customers. With the spotlight turned in the direction of new products, the attention on older products wanes to the extent that they are commercially withdrawn. The same cycle continues on the new products, which are soon replaced with newer products. As products realized through projects, it becomes difficult to manage and understand their lifecycle once the projects have been complete. Nevertheless changes are made to the product through other projects, such as projects for regulatory compliance or technology migrations, or through individual changes to add or modify features of the product. The product's contribution to the company's profitability and brand is poorly understood as the spotlight moves to the next big product launch.

Therefore when products first conceived, there should be notion of how they will be initially realized, launched, grown, operated, and their criteria for removal. Product managers should be assigned to products from the moment of their conception, and each Product Manager should have a fixed number of products in their portfolio. Products should contribute to the profitability of the company throughout their operational lifecycle, and their history of change and adaptation while being in operation should be easily visible and accessible.

A Few Skilled Gardeners should keep products in Constant Circulation through Pruning Products, Clearing Withering Products, evolving them on the basis of Recast Moulds, and identifying stable characteristics as Classic Lines. The Few Skilled Gardeners constantly create Product Markers for each Product in their care to make their characteristics and performance transparent to everyone within the Fortified Town.

## Customers create their Own



*Customers can create their own Products by combining Product Elements themselves so that no two Customers have the exact same Product.*

to be continued...

to be continued...

to be continued...



## Decide by Matching



*Equate supply and demand with matching. Qualify customers upfront through their characteristics, and matching them with product characteristics.*

Most processing structures in Telcos today, from commercial offerings to infrastructure configurations, are dominantly structured around repeatability, where standardized specifications and procedures are used to serve the needs of the many. Common denominators are obtained from market research to design products that will appeal a broad number of people, usually grouped together in Customer Segments. All customers in that segment are usually treated the same way, and all instances of customers with the same instances product are classified and processed with the same repeatable logic. For example, a customer's eligibility for a product that requires the Telco to make up-front investments like providing subsidized devices is usually accessed using a decision tree that use imperative logic and forward-chaining. The customer's credit rating, history, and other characteristics are processed to judge whether they are eligible to obtain a product. Ineligible customers for a product either find themselves at the mercy of repeating the process until they find a fit, or if they have the good fortune of dealing with a knowledgeable agent, being pointed towards a product where they can surmount the eligibility checks. Similarly, when many customers are provisioned with the same product, the same specifications are used to instantiate customer profiles, billing records, service configurations, and resource configurations. New specifications are required when the specific situation of a customer makes the standard specification unprovisionable, such as whether they live in a remote mountain top or crowded building with several lines or whether the Telco's product is operable with the customer's existing devices. As the number of specifications grow, so do the number of rules and conditions that need to be checked. Most Telcos solve this situational variety by selecting the set of situations they can service. Either by making upfront checks to decide the feasibility of providing a product or by offering simplified products that do not cater for any variety. By rejecting customers who don't fulfil a product's criteria, Telcos miss out on acquiring customers and revenue opportunities. For the few that fit the exact specifications and procedures, future changes remain within the constraints of the rigid boundaries of the product.

Therefore have several possibilities that can be a good match for any situation, and when it doubt and possible, suggest different possibilities to people so they can choose the optimal possibility. Avoid exactness and favour fuzziness, making it possible for the same product to fulfil several situations. When an entity such as a product or configuration does not fulfil an exact need, make it possible to change the entity easily until it does. Then remember the changed entity as a new possibility with correlation to the situation it helped resolve. Match a customer's demand with products that can be supplied by matching the characteristics of the customer, the customer's installed base, and the customer's preferences with the characteristics of products, the ability of the products to complement the customer's existing installed base, and the ability to meet the customer's preferences. Give customers and products to exist in their own unique combinations and situations, and possess the ability to interpret unique customer and product situations. When configuring equipment for a customer, capture the situation of the customer as accurately as possible, and suggest several alternatives based on similar location (a neighbour's configuration for a set-top box might work better than the standard one), device ecosystem (a customer who has their own media center and wants to configure it with the Telco's cloud solution is matched with the configuration used for a customer with a similar media center), consumption habits (someone who only watches mysteries in the weekend is suggested to have a subscription

that fulfils their basic needs), or financial constraints (a customer who was unable to obtain a product because of poor credit history is matched with the products provided to a customer with similar history).

Give the space for **Customers to create their Own** by using a **Language for Instances**. Rather than fixed specifications and procedures, have several possibilities where requests can be made for **One just like that One**. By **Pre-provisioning** **Frequents** and **Piecemeal Provisioning**, keep the most frequently used possibilities in memory to serve in real-time. Make it possible for **Aware Instances** to be in control of their own **Product Markers**. Create the possibility to have **Infinite Varietals**, and make products **Adaptable with Life** while keeping their **Change History**.

## Declarative Component Dependencies



*Components are aware of their relationships with other components and the Products they realize through a standard language. Products have standardized metadata to describe the components used to realize them.*

to be continued...

to be continued...

to be continued...

## Deep Roots



*Understand Deep roots for what they are. Untangle them with caution. In small portfolios, they knit the land. In a diversified portfolio, they stifle it.*

Stable products that have grown over time tend to have a lot of implementation complexity which can span several applications. Each application can have complex business rules, which in turn have complex integration with other applications. Due to the lack of automated testing, considerable manual testing effort has been expended to make these products stable. Manual processes and workarounds often exist to counter the products computational limitations, and this knowledge tends to be well-documented to ensure stable product operation.

Therefore, when new systems are designed to replace legacy systems, it is important to understand these deep roots before trying to migrate stable products. If it is not possible to understand them, let these deep roots lie, and remove them only as they are understood. Use initiatives for realizing new systems to realize new products. If stable products must be migrated, use the opportunity to evolve them into contemporary circumstances. Reduce historical essential and accidental complexity by re-evaluating the rationality that required a particular style of implementation.

By [Continuous Harvesting](#) to reap regular yields and by continuously eliminating unnecessary features by [Pruning Products](#), evolve stable products into [Classic Lines](#). Products made on the basis of out-dated molds require [Recast Molds](#) to make them more relevant to contemporary circumstances. Revisit the product's [Ringed Growth](#) to understand its historical complexity, replace them with simpler logic such as [Decide By Matching](#).

## Everyone Cares



*Customer care starts with the people most knowledgeable about the Product, and they work to spread their knowledge in different circles of expertise. It is in their own best interests to ensure that their knowledge is accessible and easy to understand in order to manage the constraints of their own capacity.*

to be continued...

to be continued...

to be continued...

## Everyone's Business



to be continued...

to be continued...

to be continued...



## Events Everywhere



*Product instances produce immutable events to reflect things that have occurred at a certain place during a particular interval of time.*

The Telecommunications industry has perhaps understood the role of Events in their architecture better than any other industry. Call Detail Records (CDRs), which are created by network elements after a telecommunications service has been utilized, have been used to record network usage for several decades. Historically, CDRs have been used to interface between Operations Support Systems (OSS) and Business Support Systems (BSS). OSS processes usually address concerns around operating the network itself, such as managing network inventories, network components (e.g. switches), and network faults. BSS processes usually address concerns around supporting customers and commercial offerings, obtaining and processing orders, processing bills, and collecting payments. OSS systems have historically been engineered to be highly distributed and fault tolerant, soft-real-time, operate non-stop, and support hot-swapping, where the system's behaviour can be changed (within limits) without stopping the system. BSS systems have historically utilized a client-server architecture utilizing transactional storage, typically with relational databases. Given the differences in the paradigms used by OSS and BSS applications, it has been historically important to create a stable and fixed data interchange format within CDRs to reflect call activity. However, people are now doing a lot more than making calls, and network systems need to be capable of capturing a wide variety of events, and BSS applications need to be able process these events. The use of normalized data models, where data is separated and categorized based on its intention or behaviour, and strongly-typed data models, where distinct data exists for each specialized purpose, will result in an extremely high scale of complexity to the point of being unsustainable.

Therefore the usage of any product should result in the creation of an event. Events should be created using de-normalized data models, where data is mostly unstructured with data elements possessing self-contained descriptors, and as weakly typed as possible, to the extent where all Events use the same object definition and contain a de-normalized data payload. The emergence of Software-Defined Networks, where network applications are built by software programs, further blur the historical distinction between BSS and OSS applications. Each application can consume Events, and create their own meaning around the Event. Events are stored in a single environment, and business intelligence and analytics can also be performed directly these events.

Many products co-exist, and their **Aware Instances** produce events that can be consumed by a wide variety of applications.

## Exotic Plants



*Introduce Disruptive Innovations in their own spaces, and fuse them in with other products cautiously.*

Stable processes for realizing and operating process are often based on continuous innovations, such as “yet another price plan”. New product ideas, and particularly disruptive innovations, that are different from existing products are often judged by the criteria used to judge existing products. Waterfall processes are tailored with templates with the assumption that one solution delivery is like another. In these cases, analysts can sense the problem through requirements, analyze the requirements based on existing delivery capabilities and behaviour in the past, and deliver solutions based on existing capabilities. Disruptive innovations, that introduce products the organization has never delivered before will require new architectures and new ways of work. Formal requirements are difficult to develop upfront as uncertainty surrounds the disruptive innovation in the beginning. Business concepts have a tendency to be vague, and rely more on competition mimicry, cherry-picking features from competitors, resorting to behavior like “make it just like the competitor”. Business value of disruptive innovations are difficult to rationalize, especially as there is no precedent for the products unlike continuous innovations. Most business value argumentation is based on competitor behaviour, which may not be resonant with brand. When judging new concepts, business impact is often judged to be “high” due to uncertainty and non-repitivity. Demands will have a tendency to be judged as immature over protracted periods of time, as they will be benchmarked to continuous innovations, leading to extensive delays. Here analysts will have to deal with greater uncertainty, requiring some extent of probing with different solution structures, then sensing the needs, and then providing analysis. In these disruptive innovations, requirements are difficult to flesh out upfront as previous experience is not helpful, and things that seem very important at the beginning of a project are not so important later. In environments with legacy systems, there is a desire to build disruptive innovations on greenfields. These greenfield environments give some initial flexibility, but if old rationality is applied to new ideas, they soon mimic legacy systems using new tools. Waste is often incurred in the beginning of realization processes for disruptive innovations by procuring all the support tooling and platforms for it from scratch.

Therefore realize disruptive and unconventional products using processes different from those used to realize conventional products. Provide the flexibility to change priorities in the development of product features based on newly obtained insights (e.g. market insights, technological learnings, and partner capabilities), unforeseen circumstances (e.g. additional complexity, market changes, etc.), and commercial opportunities (e.g. new customers, new demands from existing customers, etc.). Use these processes to realize features in products incrementally, and introduce products in their minimal form to the market as early as possible to test and understand customer response. Initially market, sell, implement, operate these products in an area of their own, and as their effect on the existing product base is better understood, merge them cautiously with existing marketing, sales, technology, and operational processes.

Disruptive innovations are **Planted in a Single Garden** within their own **Themed Area** using the most skilled amongst the **Few Skilled Gardeners** to market, sell, realize and operate the products. Provide ample **Growth Supports** when first introduced, and ensure **Ringed Growth** by offering the Product to customers as early as possible. Influence conventional product thinking by introducing the salient features of successful disruptive innovations when **Recasting Moulds**. **Repurpose Infrastructure** to realize these disruptive innovations

in order to reduce waste, but apply [Repurposed Infrastructure](#) in new roles conducive with the paradigms introduced by the disruptive innovation.



## Few Skilled Gardeners



*Have a few good gardeners to tend to products throughout their lifecycle.*

Given that products are usually launched through projects, and that projects to realize products are primarily focused on their launch, products tend to be fall to the wayside as the spotlight moves to the next product launch. The most talented individuals in the organization are allocated to the next big product launch rather than making the products of the last big product launch more efficient or being involved in the removal of inefficient products. Establishing product ownership in large organizations where each department has a different focus for the product tends to be difficult. This leads to products not realizing their true potential once launched, and a proliferation of products with marginal contributions to the organization's profitability. If Product Managers are solely responsible for administering launched products across departments, their perceived value is from the number of products they administer, rather than how actively they contribute to the profitability of the company.

Therefore selectively assign a few talented individuals as Product Managers, and establish this role as one of the most prestigious and well-compensated knowledge roles. Individuals that have strong communication skills, good market knowledge, strong technical knowledge, and operational experience are good candidates for this role. These individuals must also possess a strong network within the organization, with an understanding of how to get things done efficiently. Engage these individuals in actively realizing one or two products, operating three to five products, and retiring one to two products in large organizations simultaneously. In smaller organizations, there is greater capacity to realize, operate, and retire products given that there is less communications required for each of these activities. For the products being operated, give each Product Manager a few products that have low profitability but have great potential, a few stable and profitable products, and a few products with seemingly no hope of improving profitability. Periodically the Product Managers should exchange their most stable products with other Product Managers, which could be as often as 6 months in smaller organizations, and annually in larger organizations. The most successful product managers should be established as knowledge mentors for other product managers, and earmarked for leadership roles in the organization.

The [Three Product Champions](#), who understand the commercial, operational, and technological aspects of the product, and the [Participating Demographic](#), who are representative of the customers who would use the product, work with a select few Product Managers to [Continuously Harvest](#), [Constantly Circulate](#), and [Clear Withering Products](#). Product Managers engage in the realization of products by adequately planning their [Ringed Growth](#), ensure that newly launched products receive [Shade from Others](#) and [Growth Supports](#) within [Themed Areas](#), constantly [Pruning Products](#) in their operational portfolio, establish key criteria for the product's adoption with [Product Markers](#), and ensure that Products don't grow [Deep Roots](#). When [Exotic Plants](#) must be [Planted in the Single Garden](#), it is entrusted to one of these select few individuals. Their products can be [Changeable by Many](#) by themselves and any of the [Three Product Champions](#), giving them [Signature Aromas](#) that highlight the product's salient traits, to spread adoption and knowledge across the [Low-Walled Town Communities](#) and beyond the [Fortified Town](#) in controlled circumstances with selected partners. They frequently visit the [Town Well](#) to speak to others in the organization, and create a culture of [Specifying to Operationalize](#), which requires that the documentation around the product is just enough to make it operational. When products



don't work well, they place them in [Temporary Splints](#) until they do. They concept new ideas within [Trusted Circles](#) using [Living Prototypes](#) and [Living Data Everywhere](#), and create initiatives where the [Whole Town Participates](#).



## Fortified Town



*Competitive differentiators need to be protected. Market access to customers and suppliers through open and supervised gates.*

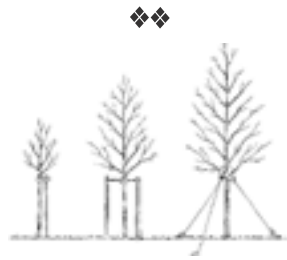
Telecommunications companies increasingly operate under intense competition with similar or comparable products. Early mover advantage, spectrum ownership, and network coverage are examples of areas where this intense competition exists. Knowledge of new products and ideas, if are leaked to the competition, would mean the loss of an early mover advantage. At the same time, telecommunications operators also have to collaborate with competitors on national policies, such as number porting, international policies, such as roaming, and industry standards, such as data exchange standards. They also tend to use common vendors and suppliers.

Therefore protect new product ideas and competitive advantages through a secure communication environment. Products that differentiate an operator from the competition should be developed internally, and products that are commoditized should be procured from common vendors. Emphasize the importance of secure communications with all employees who engage with competitors in their collaborative roles for policies and standards. In areas where resources are shared between competitors, such as shared number porting repositories or the implementation of inter-network traffic, work with each other to establish best practices. Engage with common vendors in competitive areas with caution, and ensure that vendors maintain firm segregations between customers. In areas of collaboration with competitors, engage with common vendors on the basis of best practices established through open communications. Establish and engage in communities of non-competitive telecommunications operators to exchange best practices.

Establish a secure environment within the organization, and regulate market access to customers and suppliers through supervised gates. Maintain [Low-Walled Town Communities](#) and establish [Trusted Circles](#) within this organization. Use initiatives where the [Whole Town Participates](#) to explain the importance of confidential information and keeping customer data secure at the [Town Well](#).



## Growth Supports



*Trellis, Girdle, Stake. Create just enough structures to help a product grow with time.*

New product initiatives are often used to realize previously unfulfilled initiatives. There is a desire to reuse existing experiences and existing investments, although they might not be the best way to realize a product. Products are often launched carrying a lot of expectations on their backs. In some cases, products can be used sold in ways that they weren't intended to be used, and these unanticipated applications can lead to problems in their market perception. Early failures, either because of technical teething problems, poor market adoption, or negative feedback can cast a new product in a poor light, sometimes to the extent that it is rapidly removed. Disruptive innovations, in particular, can face greater scrutiny than other products due to skeptics or champions of existing products. When products are launched, not everyone is aware of their launch, and problems with the product can often disrupt activities of individuals with no prior involvement in the product. For example, customer support agents who have not learned about a product's launch may receive calls from customers who are encountering problems with the product.

Therefore provide new products with the support they need to fuel their initial growth. Identify the salient purpose of the product and use that to base its essence. Identify what makes the product unique compared to other products in the portfolio and market, and launch products with the minimal set of functionality that fulfills its essence. Avoid grafting features from other stable products that deviate from its salient purpose and essence. Instead, try determine how the new product complements existing products, and how the most stable products, and the customers of the most stable products, can be used to support this new product. If the new product has a completely different market from conventional products, market it with products from partners that best complement its intended market. Make new product launches as visible as possible, and encourage individuals within the company to use them before their public launch. Counter skepticism about the product with clear criteria on the product's success and failure. Instruct sales, marketing, and operational staff to provide extra attention on a product that has just been launched, and ask them to anticipate unforeseen issues. If there are technical problems that impact the stability of the product, be as clear and transparent to customers about the nature of the problem and which parts of the product can still be operated. As the product stabilizes and gains market share, slowly refocus the additional support on other products.

Protect newly launched products from fierce elements by ensuring they receive have gradual and [Ringed Growth](#) and remain true to their [Signature Aromas](#) with a minimal set of features. Establish [Product Markers](#) that define their criteria for success and failure, and protect them from skeptics by giving them a chance to grow in the [Shade from Others](#) which the complement existing products well, and by placing them within [Themed Areas](#) that mark their distinctive qualities. [Exotic Plants](#) may need more shelter than others, and use the [Three Champions](#) and [Participating Demographic](#) to first launch the product within [Trusted Circles](#). Ask the [Low-walled Town Communities](#) to help the product's initial growth with the [Few Skilled Gardeners](#) keeping a close watch on the product's behavior in the first few days after its launch.



## Infinite Varietals



to be continued...

to be continued...

to be continued...

## Interdisciplinary Expertise



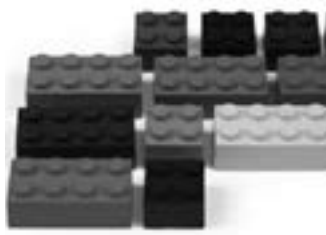
*Individuals are encouraged to develop inter-disciplinary skills. Encourage generalists who can easily specialize.*

Specialists are often required for product implementations given the technological complexity of Telecommunications environments. The Telecom industry, however, is also characterized by rapid change, and there is a tendency for specialists to desire retaining their specializations despite these changes.

Therefore focus individuals within the enterprise to build expertise in multiple disciplines. Encourage data specialists to become process specialists, and process specialists to become business specialists. Encourage everyone to perform a multitude of varied tasks.

Develop a **Few Skilled Gardeners** who can tackle almost any aspect of the product, starting from **Changeable by Many**.

## Integral Technology Stack



to be continued...

to be continued...

to be continued...

## Language for Instances



An expressive typology of entities (around 30) and possible relationship between entities (around 15). Instances can create unique expressions based on these 30 keys and 15 fingers.

## Legacy Agents



to be continued...

to be continued...

to be continued...

## Living Data Everywhere



*Lorem ipsum lifts the burden of the thinking about the practical realities of using and operating the product. Real data never looks pretty.*

Telecommunications products are used by customers in almost every aspect of their lives, whether it is at work, at home, or on the move. The same product can be used with friends, family, or colleagues in a variety of different ways. Each of these situations is reflected by data. When new concepts are first conceived, there is a tendency to ignore these diverse situations and to focus on some simplistic scenarios, often to the point of being reductionist and ignoring necessary complexity. These simplistic scenarios are easy to comprehend for the design, development, and testing of the product. The usage of processes where everything goes well (“sunny day scenarios”) and filler content (“lorem ipsum”) reflects this practice. Design agencies tend to focus on making the concept look visually appealing, sometimes at the expense of using simplistic data. For instance, consider the simple example of a drop down menu that looks great with 5 “lorem ipsum” choices, but not so nice with the actual 203 required choices. On the other hand, real production data cannot be utilized under any circumstances in design, development, or testing due to their sensitivity. Products, once they are realized, tend not to look like the early concepts because they have to cope with real data reflective of processes where many things can go wrong.

Therefore one of the biggest investments an organization can make is in creating and maintaining data for usage in design, development, and testing. This data should be inspired by real production data, but under no circumstances should it really be production data. The data should be reflective of the diverse situations faced by products, and should constantly be revised and updated based on the actual situations. The validity of the product in the organization itself should be linked to how well those who operate, modify, and manage the product manage this data. The data should be versioned so that its history of change is visible. Infrequent changes or prolonged periods without updates should indicate of this data becoming stale.

The [Few Skilled Gardeners](#) work with the [Three Product Champions](#) and the [Participating Demographic](#) create data for design, development, and testing within [Trusted Circles](#) in the [Fortified Town](#). [Product Markers](#) indicate the freshness of this data based on how frequently and consistently it has been updated. The data is used to create [Living Prototypes](#) that reflect how the product will appear through real usage.





## Living Prototypes



*Get just enough working to showcase idea. Prefer putting effort in prototypes over big specifications.*

It is difficult to comprehend the intended behaviour of a product before it is built. Some of the behaviour of a new product is often similar to existing products with some variations, such as the product's behaviour for regulatory compliance. The product's differentiating qualities are often completely new. Products are often built collectively by several departments, often using external suppliers. If the different parties involved in the product's design, development, testing, and operation do not have a common understanding of the product's intended behaviour, the product's realization will be inefficient and unstable. Documentation to help realize the product, such as requirement and solution specifications, tend to increase in volume in proportion to the complexity of the product and the number of involved parties. If the documentation becomes very lengthy, there is an increased likelihood that not everyone will read everything in the documents. Complex products with several involved parties require high volumes of communications (meetings, emails, phone calls, etc.) to ensure reliable delivery and operation, which further increases inefficiencies.

Therefore make prototypes that emulate the product that will eventually be built. Allocate up to 10 percent of the total time, budget, and resources intended for realizing the product for realizing the prototype. Build product features that are fairly obvious with static content. For instance, if the product requires users to sign in, a feature common to most products, place a static image of a login panel. Build enhancements or alterations of well-understood features that require some interaction to understand with simulated behaviour. For instance, if specialized search is part of the product's behaviour, demonstrate this ability through fixed search criteria and always return the same results. Search is universally understood in most modern organizations given the omnipresence of search tooling, and the prototype in this example would simulate how the specialized search will be different. Finally, for product features that are completely new and difficult to comprehend, build them as realistically as possible. Unlike static content and simulated behaviour, build these features so that some of the work can be reused later when the prototype is finished. For continuous innovations, where the new product is mostly like the existing products, focus the majority of the effort on static content and simulated behaviour, and only focus on working behaviour for the aspects that differentiate the new product from existing products. For disruptive innovations where the new product has very little similarity with the older products, focus mostly on simulated and working behaviour in the prototype. When building the actual product, use the prototype as a living specification. As the actual product starts being realized, decrease the effort in the prototype and increase the effort in documenting the product's accepted and stable behavior for end users. Once the product is completely realized, abandon all effort around the prototype completely, recognizing that it has served its purpose, and shifting all effort to operating the realized product and supporting documentation.

The [Three Product Champions](#) build the prototype to demonstrate the product's [Signature Aromas](#). If the product requires high confidentiality, the Prototype is only shared within [Trusted Circles](#) within the [Fortified Town](#). Given that the [Few Skilled Gardeners](#) will take the product to [Plant in a Single Garden](#), the [Three Product Champions](#) use the Prototype to demonstrate how the product will coexist with other products in [Themed Areas](#). For variations on [Classic Lines](#), the prototype should contain a higher quotient of static content and simulated behaviour. For [Exotic Plants](#), the prototype should contain a higher quotient of simulated and working behaviour. Specify to operationalize the product as its behaviour becomes accepted and stable in the realization process.

## Low-walled Town Communities



*Product development communities have low walls (chest-high) between each other in order to exchange and develop ideas easily.*

Telecom operators typically have several products that are being conceived, realized, operated, or retired simultaneously by several individuals with different areas of expertise within different departments. Given that each department tends to focus on specific aspects of a product, such as the marketing department focusing on the product's market penetration, the customer service department resolving issues customers may have with the product, and the operations department ensuring the stability of the product, there can often be some communication lapses. These communication lapses can range from inconsistent information about the product to poor reuse of individual product capabilities across the range of products offered by the Telecom operator. Features that tend to be similar between some products (e.g. price plans based around voice, messaging, and data) tend to utilize standard implementation practices, and are often fulfilled through common systems. There is also a tendency for subject matter experts to emerge based on these common practices. Dissimilar features, on the other hand, tend to be developed with specific implementations for that specific product. The marketing activities around the product tend to focus on its differences from other products, due to product's market differentiation, rather than its similarities with existing products. For instance, when Telecom Operators first launched Blackberry-based products, they were innovative products that offered a superior experience around email and internet services, and required specialized implementations by Telecom Operators. Rival devices and services soon followed for the Blackberry products, each of which had unique features that differentiating them from the others. Telecom operators soon realized that they required common means to handle IP (Internet Protocol) traffic, eventually seeing common standards and practices emerge. Furthermore, after the product has been implemented, it often emerges that the different teams could have worked together to build shared implementations, and it is these insights that evolve into standard practices. However, these insights tend to occur after the effort has been put in to develop different specialized implementations. Implementers also tend to have some pride attached to their implementations, and having an objective discussion on which implementation is more superior than the other tends to be difficult. Project managers, who are usually appointed to realize a stable product with fixed time, budgets, and resources, tend to be averse to any initiatives that could jeopardize their delivery schedule. Knowledge sharing or co-development initiatives between product realization teams are usually discouraged as they are perceived to derail individual project time-lines and scope. Individuals in these product realization teams who are familiar with the organization have the advantage of knowing the subject matter experts, but consulting these experts is then left to the initiative of these individuals. Project teams that have a high quotient of new people unfamiliar with the organization, which is often the case when using specialized vendors or new skills, who tend not to know both the subject matter experts in the organization and the existing systems complexity. Over time, each product is realized in relatively isolated silos with several areas of redundant effort and implementations. The larger the organization, the greater the likelihood that several product teams are at work simultaneously, and the greater the likelihood that they unaware of each others efforts. While it is possible to mitigate these redundancies through a strict process of governance, this either requires the individuals involved with the governance activities to have deep involvement with the individual projects or to have individual project members report their project activities to governance personnel in great detail. In either case, projects have to conform to a strict process of governance, which often introduces delays. There is a tendency for individuals

concerned with governance to focus on reuse and standards, which often lead to additional implementation considerations, additional resource capacity, and additional time. These are usually the three things of which project managers have an aversion. Finally, innovative or differentiating products are often developed under strict confidentiality to ensure that the competitive market advantage or first-mover advantage remains until the launch of the product.

Therefore create visible work-spaces within and between product-teams. Product teams should operate with the highest security clearance in the organization, which would permit them to speak to each other without fear of compromising sensitive information. Individuals responsible for product design should document their work on spaces where their work is visible to all members within and beyond the product team. Documentation should be focused on helping the product go operational. Individuals responsible for the implementation of products should open their source code and configurations for scrutiny by all product teams. Source code, configurations, prototypes, product data, and documentation are visible, searchable, and accessible by all product teams. Products are deployed using a common language for the components utilized by a product. The products themselves can coexist with other products by design, and each product advertises a set of events that other products can consume. Products react to these events, giving the possibility for no two product instances to be exactly the same. Product teams have regular rhythms for meeting during the week in common spaces, such as fixed face-to-face or virtual knowledge sharing sessions, and these regular rhythms are respected. If rhythms are missed due to some reason, this is made visible within the organization. The success of a product team is judged by how well they deliver their products and how well they share their knowledge with other teams. Product managers constantly meet other product managers to discuss product features, implementations, and practices. Product teams first take their products and test it within the organization, and are able to receive feedback from anyone within the organization. Non-product team members of the organization, such as customer support and sales staff, have access to a common product knowledge base. This knowledge base contains information about how to use the product, usage of the product, common issues, and issue resolutions. Almost everyone in the organization of the Telecom operator should have access to this information. Sensitive commercial information should be avoided in this knowledge base. Finally, parts of this knowledge base should be opened up to customers so they can provide direct feedback about the usage and experience of the product.

Product teams operate within [Trusted Circles](#) where they can freely share information with each other within the [Fortified Town](#). By [Specifying to Operationalize](#), the [Three Champions](#) and the [Few Skilled Gardeners](#) create and encourage others to create documentation that helps both sustain the product itself, and other products so that [Many Products co-exist](#). [Living Prototypes](#) and [Living Data Everywhere](#) created by product teams are visible to other product teams, as well as Source Code, Configurations, and Documentation. Using a common language for Declarative component dependencies, teams are able to view Components, understand their usage, and discover opportunities for reuse. Product teams meet at the [Town Well](#) to discuss their products, and when their products are ready, they share them first with people within the organization so that the [Whole Town Participates](#). The [Participating Demographic](#) and the [Three Champions](#) lead efforts for usage adoption within the organization, encouraging all participants and users to utilize the Product and share experience and knowledge on the Productpedia.

## Many Products Co-exist



*Products are aware of each other and coexist together. Products broadcast events to other Products.*

Among the first consumer-oriented telecommunications products were voice services to connect 2 people and telegraph services to send data over Morse Code, which remained a stable commercial product offering for several decades. The emergence of mobile and internet technologies enabled new product offerings like messaging, multi-party communications, and internet services starting from the 1980s. The relatively recent emergence of smart devices has created significant intelligence at the edge of telecommunications networks. Almost every aspect of life including work, entertainment, health, travel, finance, and insurance can be accessed through multiple devices in the form of services. Individuals utilize several of these services over the internet, potentially reducing telecommunications operators to the role of “the dumb bitpipe”. This is not dissimilar to the role Internet Service Providers have been playing, unable to establish few services in the lives of customers other than providing connectivity. Increasingly, users tend to have several applications and services to address the different aspects of their connected life. However, these applications and services are rarely connected, and users must be able to cope with their fragmented nature in order to utilize them effectively. Users often buy the applications, services offered by these applications, and content either directly from the device ecosystem or from the operating system’s ecosystem. Telecom operators have recognized the value of including some of these applications, services, and content into their existing offerings, such as fixed price tariffs for music (e.g. Spotify) and movie consumption along with traditional products for voice, data, and messaging. The addition of these applications, services, and content with traditional products introduces greater complexity, especially if each of these products must be able to inter-operate with each other. Most Telecom Operators are already weighed down by the number of existing products they have from their traditional portfolio, and introducing all of these applications, services, and content as additional product offerings introduces exponentially greater complexity. Traditional integration methods to make these products inter-operable require that each Product be conscious of the data model applied by other Products, eventually requiring a Canonical Data Model in Middleware to mediate between all of the Products. This introduces even greater complexity at the center, requiring a set of data interchange interfaces between Products to the Canonical Data Model, and another set of data interchange interfaces between the Canonical Data Model to the individual Products. Furthermore, the introduction of new concepts requires the re-modeling of data interfaces at each interchange. The center gets very heavy and complex, in accordance to the law of requisite variety. This large scale of complexity introduces even greater complexity into the existing Product landscape making User Experiences increasingly error prone, expensive, and inflexible.

Therefore some Products need to know about other Products, and when they need to know about other Products, they create a perspective that is unique to that Product. For instance, an Insurance Product will want to know about your Device Portfolio (managed in a Device Portfolio App or Device Ecosystem), and utilize this information to make insurance offers to insure these Devices. The application managing the Device Portfolio only needs to provide information about the Device, and does not have any knowledge of the insurance risk of these Devices. Products are able to inter-operate with each other by design through Events, which contain information the usage of the Product in a de-normalized structure. Each of these Events are persisted immutably, and made available to other Products used by the same User. Products consume Event information in a de-normalized structure, and



interpret the data in their own way. Individual Product Developers decide what kind of Products they want to inter-operate with and communicate with other Product Developers. The eco-system where the Products exist are responsible for taking Events and reliably persisting them. Products regularly listen for new Events, and consume them if they are pertinent. Each Product decides how to process an Event in its own way, creating meaning, reactions, and actions to the Events in their own interpretive manner.

The [Three Product Champions](#) exhibit their products at the [Town Well](#), advertising the information their product can offer other Products. They consult the [Few Skilled Gardeners](#) about the sort of information other products in which other products might be interested. [Aware and Integral Instances](#) generate events which are persisted in a discoverable event store. By using [Living Data Everywhere](#), the co-existence of products can be tested before they are launched.



## Meaningful Names and Identities



*Using Mobile Subscriber Identities (MSISDN) or Directory Numbers will change with the emergence of IP technology, giving way to meaningful identities.*

An individual's mobile number, also technically known as their MSISDN (Mobile Station International Subscriber Directory Number), or their Directory Number, which is referred to as their landline numbers or PSTN numbers, have been stable means to identify lines and users. These methods of identifying users are currently based on traditional communications technology like GSM and PSTN. There is a tendency for these identities to play a pivotal role in Customer Care systems, with distinction between Users (people who use Products) and Customers (people who pay for Product usage) being blurred. As world populations, the number of users, and the number of uses of Telecom services increase, the shortage of numbers within current ranges becomes restrictive for new business models. If numbers get too large, they become harder to remember, and increasingly the role of Phone Books on Devices have become more important to remember large volumes of numbers in a meaningful way. When Customers switch operators, the porting of line identities (number porting) tends to be a cumbersome process (especially in Europe), although regulators increasingly want to make them more efficient. As IP (Internet Protocol) technology replaces traditional communications technology, these legacy identities will lose their pivotal role. In an all IP technology stack, IP addresses rather than MSISDNs and Directory Numbers will be used to identify the line. While this presents an opportunity to revise the existing practices around line identities in Network Technology, Customer Care, Billing, and Number Porting, the deep entrenchment of these concepts within existing practices makes change very difficult. Users who have grown accustomed to their "telephone number" as a part of their identity may also want to retain that identity while they get used to other form of emergent identities. Finally, the emergence "the internet of things" or "

Therefore make a distinction between absolute identities that remain integral over a period of time and referential identities which are easy for users to remember. IP addresses are an example of absolute identity, and usernames, MSISDNs, and Subscriber Directory Numbers are refer to the IP address. Referential identities can be used to contact individuals, and network systems resolve referential identities to absolute identities. Existing logic within Customer Care, Billing, and Number Porting that utilize the legacy identities can remain, and be transitioned as the referential identities become less and less significant.

The many that products co-exist are linked through a common identity, and ensure these common identities don't grow [Deep Roots](#). The [Aware and Integral Instances](#) exist with absolute and referential identities that utilize the [Language for Product Instances](#). The [Product Markers](#) refer to individual Instances of Products, continuously measuring their performance.

# Multi-criteria Directories



to be continued...

to be continued...

to be continued...





## Nurture Keystones



to be continued...

to be continued...

to be continued...

## One just like that One



to be continued...

to be continued...

to be continued...

## Own Base Currency



*Operators have their own Base Currency, and Customers are compensated with a Telco's own Products. Conversions to 3rd party products, such as Retail Loyalty Points, is done from the Base Currency.*

Telcos have traditionally combined access-dependent products such as voice, messaging, and data around rigid subscriptions with fixed durations in contractual arrangements. Attention has traditionally focused on two critical moments in a customer's lifecycle, acquisition and retention. In markets where Telco product ownership has reached saturation point, in that anyone who is a potential Telco customer is already with a Telco provider, operators are focused primarily on retaining customers whose contracts are about to run out (retention) while trying to attract customers leaving other providers (acquisition). Subsequently, the end-dates of customer's contracts are highly sensitive information, and cases where this information is leaked to rival operators is treated with the greatest severity. Some operators go to the extent of obfuscating the end-date from their own customers, which usually adds to the distrust. Operators try and entice customers to stay with generic retention offers towards the end of their contract. There is usually very little focus on the customer's lifecycle after they have been acquired and their contracts are about to run out. The customer's individual experience with the brand is poorly understood, if at all, and the few interaction moments a Telco has with their Customers tends to be over bill disputes or service issues. Although many Telcos have loyalty programs based on usage, often retrospectively calculated from their monthly bills, which customers are able to redeem through loyalty partners. Most customers with contracts also have direct-debit arrangements, where the monthly dues are automatically credited to the Telco's account. Customers are increasingly unlikely to read the paper invoice posted to their homes that may contain their loyalty rewards, and many operators provide an option to only receive electronic bills. Loyalty programs are usually operated with retail or travel partners, and access to these programs is usually only through a customer service channel, which is typically used only when there are issues. As loyalty rewards are usually granted through partners, the currency of partners is used to reward customers, such as loyalty points from an Airline or Retail chain. As a consequence, most customers are unaware of their loyalty rewards, and these loyalty rewards are rarely used to retain customers. Customers who are aware of their loyalty rewards spend them with retail or travel companies. While these companies benefit from enriching their customer profiling knowledge, this knowledge is rarely shared with the Telco operator. Historical attempts to reward customers in terms of their own products of voice, messaging, and data has also been difficult as it has been difficult to redeem rewards as consumption occurs. Rather, reward redemption in terms of their own products have historically occurred with the generation of the bill, making the experience between consumption and redemption too far apart. With an all Internet Protocol world and the prevalence of Smart phones, customers are increasingly using their data connectivity to consume a variety of access-independent products, ranging from media to augmented reality. The consumption of these products are usually other providers and not the Telco, where operators are only used for access to these products. Most operators are not able to continue relationships with customers after they decide to leave for another provider, as the termination of their contract brings an end to all relationships.

Therefore offer all loyalty rewards using the Telco's own base currency, and correlate the base currency to the Telco's own products. Offer variable exchange rates to the products of others within moments of the consumption. Customers are able to view the balance of their rewards across multiple devices, and options for their redemption are always visible with the balance. Customers are able to redeem their rewards to any access-based products such as



media within moments, making it possible to redeem rewards gathered making phone calls to watch a free movie or a cheap international call. Offer products contextually based on a customer's location, time-of-day, and social occasion. Exchange rates are variably set based on demand, network capacity, and availability. Customers renewing their subscriptions are provided relatively a high amount of rewards, and customers who decide to leave retain their rewards, either to spend on the access-independent products of the operator or with partners. Companies that have their Telco contracts for their employees with an operator have the option to keep the rewards themselves or distribute them to their employees. The Telco's own base currency is accepted as a primary means of reward and compensation across all of Telco's sales channels. Customer referrals, where they ask a friend or family to join the operator, is also rewarded with the Telco's own currency.

Use the [Language for Product Instances](#) to understand a customer's situation and [Change History](#), and ensure that [Everyone Cares](#) about enriching a customer's experience with the operator. [Nurture Keystones](#) by rewarding them with the [Own Base Currency](#), and permit Customers to ask for [One just like that One](#) anytime in their relationship with the Operator by being [Adaptable with Life](#). When customers choose to Port Providers, let them go with grace and let them retain their rewards for spending with the operator they are leaving or others.

## Participating Demographic



*The demographic develops the product. If you are developing a product for senior citizens, get senior citizens on board as product managers.*

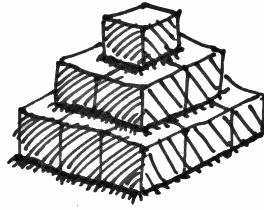
Product designers who make products they themselves would not use often make irrelevant products for their target market.

Therefore the product design process needs involvement from individuals who would actually use the product.

Shortly after the [Three Product Champions](#) get together, they work to find [Skilled Gardeners](#) and individuals who are representative of the target market ([Participating Demographic](#)) and include them within [Trusted Circles](#). If they do not work for the Operator, they are provided an opportunity work within the [Fortified Town](#). They give continuous usage feedback with the Product to the [Three Champions](#), and participate with as many products that are relevant to their Demographic as possible. As the product nears launch, the members of the [Participating Demographic](#) meet frequently at the [Town Well](#) to discuss their experiences within the product within people within the [Fortified Town](#).



## Piecemeal Provisioning



to be continued...

to be continued...

to be continued...

## Plant in a Single Garden



*Virtual products can seemingly possess infinite space, but in reality only have finite capacity. Represent all products as occupying a single space.*

The virtual nature of telecommunications products lends the impression that there is infinite capacity for products, but there exists finite capacity in individuals and systems for development, operational, and processing capacity. Products have a tendency to proliferate both in terms of diversity and volume, given this illusion of infinite space. Due to the diversity of the products, consumers have inconsistent experiences of the brand. Due to the volume of products, it becomes difficult for consumers to differentiate between products to the extent that products start competing with other products. Also due to the high volume of products, it also becomes unclear how all of the products complement each other.

Therefore use a finite visual space to represent all of the offered products. This space can only hold so many products at a time, and when this finite space becomes full, the order and placement of products must be re-arranged or products must be removed. Products can be arranged in this finite space based themes such as their ability to complement other products in common spaces or similar characteristics. As the arrangements stabilize, new products are always are placed this finite space in terms of the existing products.

Display all of the offered products using a strong visual metaphor representing a finite space at the [Town Well](#). Through [Constant Circulation](#) and by [Clearing Withering](#) products, introduce a few new products while removing a few old products. Create [Themed Areas](#) within this finite space to group Products together where young products have [Growth Supports](#) and receive [Shade from Others](#), and permit their collective [Signature Aromas](#) to spread within the [Low-walled Town Communities](#). The latest [Recast Moulds](#) should exist in a place of prominence in this visual and finite space.



## Pre-provisioned Frequents



to be continued...

to be continued...

to be continued...

## Product Markers



*When is a product doing well? When is it doing poorly? When conceiving the product, develop the criteria for its decommissioning it at the onset.*

## Productpedia



*Create a common knowledge base of the product, searchable by everyone involved with the product.*

to be continued...

to be continued...

to be continued...

## Pruning Products



*Constantly trim products to remain true to their purpose. Avoid features for the sake of features.*

to be continued...

to be continued...

to be continued...



## Provider Porting

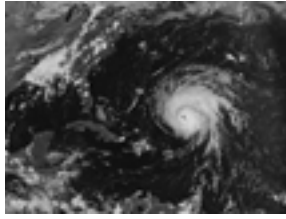


to be continued...

to be continued...

to be continued...

## Real Time Offers



to be continued...

to be continued...

to be continued...



## Recast Moulds



*Periodically, there should be a product which is leading the organization's focus. Recast this product every annually in large organizations, and every 6 months in medium sized organizations.*

to be continued...

to be continued...

to be continued...



## Repurpose Infrastructure



*Reuse old hardware for new purposes. Reuse systems for new purposes. Unanticipated applications.*

## Ringed Growth



*Launch the product with the minimal set of functionality required to operate it in the market.  
Grow it cautiously. Don't overburden it with features that never get used.*

## Rhythmic Sensemaking Spaces



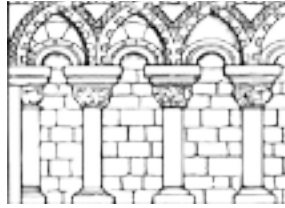
to be continued...

to be continued...

to be continued...



## Several Prototypical Configurations



to be continued...

to be continued...

to be continued...

## Shade from Others



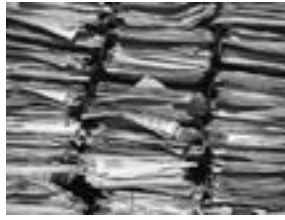
*Give new products the room to grow in the middle of bigger products.*

## Signature Aromas



*Give each product a unique aroma. Permit the sweet and unique scent of the product being developed to drift beyond the walls of the product development team.*

## Specify to Operationalize



*Specify enough to build a prototype, and then stop. Grow the prototype enough to build a system, and then stop.*

to be continued...

to be continued...

to be continued...

## Temporary Splints



*If an operating product has defects, create splints that allow its continued operation while the product is being repaired and fixed. Splints are visible to users. Caution of malicious users.*



## Themed Areas



*Plant products that complement each other in the same space. Understand how products are conflicting and congruent with each other.*

to be continued...

to be continued...

to be continued...

## Three Product Champions



*A product needs to be championed by at least 3 people in order to accepted as a candidate for development.*

Poorly conceived products are difficult to implement, and poorly implemented products are difficult to operate.

A product needs to be championed by at least 3 people in order to accepted as a candidate for development. One person close to the market with knowledge of how to market and sell the product, one person close to enabling technology with knowledge of how to realize and implement the product, and one person close to operations with knowledge of how to provide customer service, support, and stability to the product.

Working with the [Participating Demographic](#) and a [Few Skilled Gardeners](#), the three individuals concept, prototype, develop, and operate the product.

## Town Well



*A single source of nourishment where town members return continuously throughout the day.*

## Trusted Circle



*Product development communities have low walls between each other in order to exchange and develop ideas easily. Have other product communities test products together.*

## Usage View as it Occurs



to be continued...

to be continued...

to be continued...



## Visible Assists



to be continued...

to be continued...

to be continued...

## Whole Town Participates



*Create common targets for the whole company where all individual can participate in product community. The first customers of a product are internal staff. Whole town tests together.*

Large organizations are divided into departments, which is necessary because of many reasons... A common instrument of managing goals for individuals is targets, the fulfilment of which reaps financial rewards for individuals. Top down, strategy broken down, little goals. Departments get their own goals, but Individuals only fulfil things within their targets. When Targets for an individual are set by historic precedents, and rely on repeating the past effectively. Targets are translated in terms of a department. Technology department focused on transformation while Marketing department focused on maximizing on variations of existing products, while Sales department focused on continuous innovations on existing products.

Therefore, create common targets for the whole company where all individual can participate in product community. The first customers of a product are internal staff.

The [Three Champions](#) and the [Participating Demographic](#) spread word about the new product at the [Town Well](#) across [Low-walled Town Communities](#) but within the [Fortified Town](#). [Growth Supports](#) are provided to individuals as incentives to help test the product, and the [Signature Aromas](#) of the product emerge and drift across the [Low-walled Town Communities](#). The product's emerging limitations and defects as it is with the product are visible to all participants at all times and if necessary put in a [Temporary Splint](#).