Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Mindmap books

Make a mindmap of books you read, to expose the structure of the ideas within.

Also: **The Feynman Method**.

Related

Be approximately right
Low floor, wide walls,…
Brick pencil
Intuition is pattern re…
3 is the magic number
Learn to learn
We mistake stories f…
4 ways to augment t…
OODA loop
Questions are places…
Pensees
Capture, organize, s…

Places to intervene in a system

From Leverage Points: Places to Intervene in a System, by Donella Meadows:

> PLACES TO INTERVENE IN A SYSTEM
> (in increasing order of effectiveness)
>
> - Constants, parameters, numbers (such as subsidies, taxes, standards).
> - The sizes of buffers and other stabilizing stocks, relative to their flows.
> - The structure of material stocks and flows (such as transport networks, population age structures).
> - The lengths of delays, relative to the rate of system change.
> - The strength of negative feedback loops, relative to the impacts they are trying to correct against.
> - The gain around driving positive feedback loops.
> - The structure of information flows (who does and does not have access to information).
> - The rules of the system (such as incentives, punishments, constraints).
> - The power to add, change, evolve, or self-organize system structure.
> - The goals of the system.

- The mindset or paradigm out of which the system — its goals, structure, rules, delays, parameters — arises.
- The power to transcend paradigms.

Backlinks

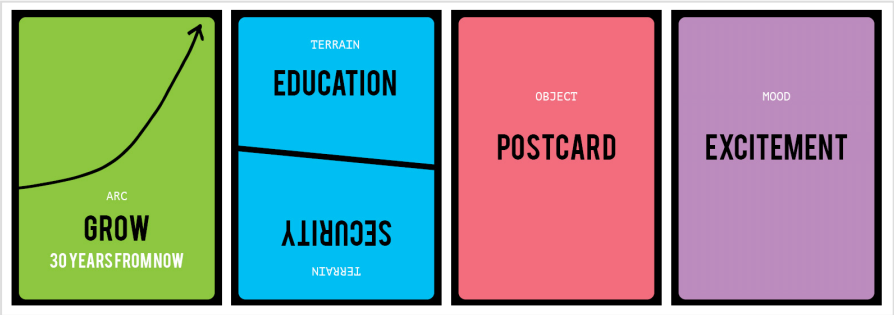
TIMN

Related

Tools over technique	Second System Syn…	Modularize later
Low floor, wide walls,…	Start with a toy	Evolution
Permissionless creat…	Manifesto	Measure it
4 ways to augment t…	Cities are like compo…	Generative Grammar

Thing from the Future

Thing from the Future is a deck of cards for **scenario planning**. It's a game about peeking into a future and pulling objects from that future into the present. Cards come in 4 flavors:



- Arc cards describe a future trajectory (grow / collapse / discipline / transform) and a timeframe (10 years, 100 years, etc)
- Terrain cards describe a context ("Education" / "Security", etc)
- Object cards define the thing from this future.
- Mood cards ("excitement", "dread") define an emotional inflection.

The job of players is to synthesize these prompts into a future scenario, then create an object from this future.

Related

Cities outlive civilizati…
A Pattern Language
Human scale
The Art of Game Des…
Aurora
Procedural paintbrush

Desirability, feasibilit…
Lindy effect

Understanding Comics
Create choices, mak…

Recreate a piece of art
Scenario planning

Gimmick Book

Pulp sci-fi author Alfred Bester kept a book of storytelling tricks.

> All I ever wanted was to be a great storytelling pitchman, which is why I collected the tricky devices and means which are entered in this Gimmick Book.

Why?

> Back in the 18th and 19th centuries literati would have called it a Commonplace Book, 'a book in which noteworthy passages, poems, comments, etc. are written.' Literature was a diversion for them. For me it's a livelihood and this Gimmick Book is my toolbox.

Keep your own Gimmick Book
I am a believer in the power of the Gimmick Book. Whatever you do, capture it. Write it all down. These scraps are your livelihood.

- Gimmicks can be the beginnings of **design patterns**.
- Capturing them is the first step in **capture, organize, synthesize**.

The Waste Books
Bester wasn't the only one. Georg Christoph Lichtenberg was an 18th c astronomer, mathematitcian, and physicist who collected his stray notes and thoughts in The Waste Books.

Apparently Tolstoy, Nietzsche, and Wittgenstein were fans.

Pascal's Pensees
Sometimes the Gimmick Book itself becomes valuable, as with Pascal's **Pensees**.

Related

Manifesto	Jack principles	Generalized problem…
Places to intervene i…	OODA loop	Brick pencil
Human scale	Thing from the Future	4 ways to augment t…
Recreate a piece of art	A Pattern Language	Architects vs garden…

Networked things

Tesla recently announced self-driving as an add-on feature for future models. Buried in the legal jargon is this: You're not allowed to use your self-driving Tesla "for revenue purposes". Who owns this car?

On Oct 21, 2016 a massive botnet took down DNS service Dyn, shutting down access to many large sites in North America.

The botnet was mostly made up of zombie webcams infected by Mirai -- viral malware designed to turn IOT devices into a DDOS army.

On Oct 24, the webcam manufacturer Hangzhou Xiongmai, recalled the insecure devices that had made up a large part of the botnet. But this single webcam model is hardly the root cause.

> Oh, hey, my solar panels have an open port that requires auth against a hard-coded password that *advertises the allowable usernames*. 😬 pic.twitter.com/cvGG3gcVBf
>
> — Waldo Jaquith (@waldojaquith) October 22, 2016

We live in a future where you can buy a $16 Linux computer the size of a stick of gum. Solar panels, webcams, cars, toasters, candles... Everything will be a little smart. Everything will be hackable.

Some questions:

- Who will own these smart things?
- What does ownership mean when a hacker can compromise them?
- Is "ownership" a realistic goal for a networked thing? Is it desirable?
- When everything we own is a little bit smart, will all those smart things have overlapping and competing legal agreements?
- If so, who decides what is fair?
- Who will be responsible when things go wrong?
- How will we know what EULAs we are breaking and what EULAs we are keeping?
- With thousands of competing EULAs, how will conflicts be enforced? Automated settlements?

- A16Z: The End of Ownership.
- Kevin Kelly: Sharing is better than owning and platform trumps product.
- The battle for the customer interface: "Uber, the world's largest taxi company, owns no vehicles. Facebook, the world's most popular media owner, creates no content. Alibaba, the most valuable retailer, has no inventory. And Airbnb, the world's largest accommodation provider, owns no real estate.".
- The common thread running through the sharing economy... there's nothing on the balance sheet.

Related

Scenario planning	Second System Syn…	Prototyping
Not Rocket Science …	Hash string UX	Science is the only n…
Modularize later	Aurora	Judo move
General-purpose lan…	Pace layers	Conformability

Goodhart's law

Goodhart's Law:

> When a measure becomes a target, it ceases to be a good measure.

Goodhart's Law applies to feedback system with humans in the loop. It does not apply to feed-forward systems.

Any system including people is a kind of feedback system, because people are smart. We are **agents**. We observe outcomes and change behavior in response to system pressures. **Evolution**! The measure will cease to be a good measure when participants become aware of it and optimize for the measure, rather than for the outcome that the measure was intended to proxy.

Evolution

Evolution is a behavior that emerges in any system that has:
Mutation Heredity Selection
Evolutionary systems often generate unexpected solutions

Measuring something? Be careful how you **measure it**.

Backlinks

General-purpose lan… Pournelle's Iron Law …

Related

Punctuated equilibrium
Composable alphabets
Design patterns
Job-to-be-done
Containment can lea…
Generalized problem…
Language is a Lamar…
100 Users
Value, scale, time
Verbs that can act on…
Amara's Law
Pournelle's Iron Law …

Be approximately right

> Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.

— John W Turkey

Consider the value of an answer along 2 axes: accuracy and precision.

- High accuracy, low precision — "Pi is 3.14"
- High precision, low accuracy — "Pi is 5.2303948911230394912031 57"

See also **intuition is pattern recognition**, **learn to learn**.

Backlinks

The Feynman method

Related

The Feynman method
OODA loop
Pensees
Gimmick Book
Slowly, then all at once
Solvitur ambulando

Job-to-be-done
Recreate a piece of art
Modularize later
Brick pencil
Manifesto
Conformability

Magic circle

The Magic Circle is a metaphor for thinking about the time/space in which a game is played.

- The magic circle has boundaries of space and time. A soccer pitch is a different place before vs during the game.
- When players enter the magic circle, the game mediates their interactions.
- Inside the magic circle, specialized meanings are given to objects and actions.
- The magic circle is a temporary world-within-the-world.
- Entering the magic circle requires suspending disbelief.

Questions:

- How can I mark boundaries?
- How can I arrange objects/people/space to facilitate play?
- How can I mark special meanings and events?
- How can I create a temporary world, separate from ordinary reality? How should I set the scene?
- How can I embrace inefficiency and suspend disbelief?

See book Rules of Play for more on this topic.

Related

Generative Grammar
The Art of Game Des…
Manifesto
Villain, Team, System
Games with more th…
Time consciousness
Verbs that can act on…
Hyperreality

Build to think

It's best to explore new opportunity spaces through **prototyping**. To try to model an opportunity space in your head is to explore that opportunity space in your imagination. You'll only find what you already know.

Prototyping is the process of following loose threads in the Gordian Knot. You'll learn more about the space and spot opportunities you didn't know existed. You'll also uncover risks in places you hadn't expected.

> I'm figuring this out as I go. One's ability to articulate an idea always lags behind the understanding of the idea, and the understanding of an idea often lags behind the embodiment in which it is first given life. It can take a surprising amount of time to come to understand what a prototype is trying to "say" and longer still to say it oneself.

— Brett Victor

Backlinks

Bootstrapping Solvitur ambulando Create choices, mak…

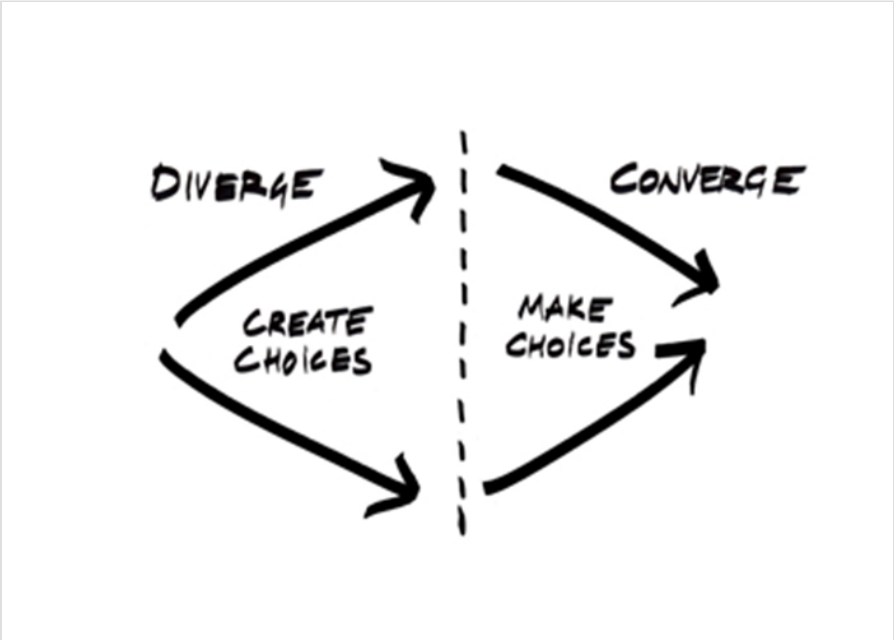
Related

Desirability, feasibilit… Gimmick Book Understanding Comics

Pace layers
Recreate a piece of art
Invisible Cities
Architects vs garden…
Judo move
A Pattern Language
Brainstorming
Human scale
Thing from the Future

Create choices, make choices

The creative process in a nutshell:



1. Create choices. This is a generative frame of mind. Suspend judgement and analysis. Breadth over depth. You're building up raw material. **Brainstorming** is a good tool here.
2. Make choices: trim, edit, modularize, winnow, polish. This is a reductive frame of mind. Your goal is to take the raw material and form it into a polished concept.

This process is also sometimes called blue sky, brass tacks:

- Blue sky: what's the best outcome we can imagine? No limits, no constraints.

- Brass tacks: how do we get there? What are our constraints? What resources and tools do we have available? This is were you start planning, using methods like **Charlie Kindel's 5 Ps**.

These 2 stages also map to problem-solving methodologies used by design and engineering.

Design uses a range of strategies from both stages. Intuition, and creative strategies like moodboards, form finding, and **brainstorming** are used to create choices. Iteration, design reviews, critique are used to make choices.

Production engineering is typically a process of making choices for implementation. It uses a strategy of modularizing problems — breaking problems into smaller parts, then solving each part. It can also be exploratory. **Prototyping** can be a process of creating choices when you take the **build to think** approach.

It's essential to know which stage of the process you're in, and whether to use a generative or a critical process. If you winnow before you sow, you'll have nothing to show.

See also: **capture, organize, synthesize**.

Backlinks

Generalized problem…

Related

The Feynman method	Mindmap books	3 is the magic number
Desirability, feasibilit…	Brick pencil	Be approximately right
TUI	Build to think	4 ways to augment t…
OODA loop	The Art of Game Des…	Gimmick Book

Amara's Law

> We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run.

Why? According to **Clark's 3 laws**, failure of nerve and failure of imagination.

Backlinks

Fundamental Needs

Related

Networked things
Modularize later
Manifesto
Cities are like compo…
Tools over technique
Cities outlive civilizati…
Pace layers
Games with more th…
Science is the only n…
Conformability
Value, scale, time
Bootstrapping

Conversational UI

Lots of what passes for an AI revolution is actually a change in interaction model, enabled by machine learning that has gotten "good enough" to suggest a single answer to a query.

This is an important shift. As compute becomes smaller and cheaper, it is infecting everyday objects that we don't think of as "computers". In the future, everything will be a little smart. Conversational UI will be an important tool for interacting with these smart things, because many will be too small to have screens.

Advantages

- Works anywhere — scales from screenless devices to mobile to Desktop.
- Conversations are an anthropomorphic way to interact with computers.
- Conversational UI can seamlessly blend interactions between people and artificial intelligence agents.
- Messages, conversations and notifications have heavy conceptual overlap.
- Search, conversations, notifications, messages and cards — these things have a strong overlap in terms of jobs and features.
    - This is why WeChat is able to expand into a platform.

Risks and Challenges

- The rule of thumb for machine learning is that 80% success rates are easy, but the next 10%-15% gain is hard.
- Failure hurts. How often are you quietly exasperated by Siri? When users feel punished by failure, they stop exploring and stick to the beaten path.
- Where AI falls short, careful UI can patch up the differences. This is where design patterns come in. WeChat does this very well, with structured responses.

Design Patterns

- The Cooperative Principle: we trust that others we converse with are being cooperative, and will contribute usefully to the conversation. Avoid breaking this deal at all costs.
- Be As Smart as A Puppy. If the computer is dumb, make it charming.
- Keep conversations structured. If you can't have 100% accuracy, narrow the problem space with careful UI design.
- The **Jack Principles** of conversational UI
- **Augmented conversations** — chatbots can seamlessly switch between bot and human. This is already a thing on WeChat.

Stray thoughts

- When voice assistants get good enough, every business will have a programmable API, because every business has a phone.
- What happens when we have many of these bots, conversing with each other?

Resources

- 2015 Ben Evans: The State of Messaging
- 2015 Interconnected: Conversational UIs

- 2015 Interconnected: More on Conversational UIs
- 2015 Alistapart on Conversational UI
- 2015 Alistapart Designing for Conversational UI
- 2015 Futures of text
- 2015 [Wired: The future of UI design? Old-school text message] (http://www.wired.com/2015/06/- future-ui-design-old-school-text-messages/)
- 2014 Chinese mobile app trends. The future of the web is here, it's just not evenly distributed.
- 2003 Matt Webb on Conversational UI

Related

Augmented convers…	Recreate a piece of art	Brainstorming
Desirability, feasibilit…	Gimmick Book	Human scale tools
The Art of Game Des…	Invisible Cities	Brick pencil
TUI	Capture, organize, s…	Build to think

Not Rocket Science Rule

The Not Rocket Science Rule Of Software Engineering:

> Automatically maintain a repository of code that always passes all the tests.

The Rust language does this with automatic pull request review bots. No code is landed unless it passes tests. Bots do all the code landing so there can't be mistakes.

- Not Rocket Science Rule Of Software Engineering
- Rust infrastructure can be your infrastructure

Also: Software development, Devops

Backlinks

Project management…

Related

Judo move
Modularize later
Networked things
Second System Syn…
USLICE
Hash string UX
General-purpose lan…
Prototyping

Villain, Team, System

> In a game, there should always be some element that works against player success. This role is often taken by a villain, a competing player or team, or may be embodied in the game system as a whole.
>
> —Rules of Play

Where is the conflict coming from? (Villain? Team? System? Myself?)

Also: **Games with more than one player**, **Hero's Journey**, **story arc**.

Related

Generative Grammar
Magic circle
Verbs that can act on…
The Art of Game Des…
Games with more th…

Solvitur ambulando

> Solvitur ambulando / "it is solved by walking". — Diogenes the Cynic, responding to Zeno's paradox.

Solve it by **prototyping**. Solve it through experiment. **Build to think**.

> A creature didn’t think in order to move: it just moved, and by moving it discovered the world that then formed the content of its thoughts. — Andy Clark

---

> Solvitur ambulando. Truth is process, not object. — Iain McGilchrist, The Master and his Emissary

Backlinks

OODA loop

Related

Pace layers

Pace layers are a model for thinking about change, developed by Stewart Brand.

Each layer acts as a platform for the layer above it, and moves more slowly, else it wouldn't act as a platform.

When exploring a problem space, ask

- What is changing quickly?
- What is not changing?

You have identified a pivot point.

Hunch — large-scale discontinuities in one layer will have dramatic effects on layers above, but attenuated effects on layers beneath.

Also: **scenario planning**.

Backlinks

Slowly, then all at once
Cities outlive civilizati…
Lindy effect
Conformability

Related

How human systems…	Composable alphabets	Aurora
Cities outlive civilizati…	Cities are like compo…	Architects vs garden…
Thing from the Future	Building Block Hypot…	Markets do not make…
Gimmick Book	Gall's Law	Recreate a piece of art

Manifesto

Manifestos are statements of absolute belief. They offer no provision for measuring outcomes and course-correcting. Manifestos can't change their mind.

A manifesto is a feed-forward system.

Related

100 Users
4 ways to augment t…
Job-to-be-done
OODA loop
Start with a toy
Goodhart's law
Lindy effect
Gall's Law
Games with more th…
Modularize later
Be approximately right
Places to intervene i…

The Art of Game Design

Games are an extreme case of interaction design, making this book a complete introduction to IXD and UX.

Schell structures his book around thinking "lenses" — multiple perspectives you can try on when evaluating a design. Each lens comes with an introduction and questions that can be used as design prompts.

Along the way, the book also covers the basics of agile development, project management, team structure, creative brainstorming, how to get past creative block... it's quite a tour.

Backlinks

Verbs that can act on…

Related

Harvesting the Biosp…
Desirability, feasibilit…
Augmented convers…
TUI
Extrastatecraft
A Case for Climate E…
Understanding Comics
Thing from the Future
Generative Grammar
Robotics Primer
Invisible Cities
Brainstorming

Capture, organize, synthesize

Takeaway from a Mozilla user study studying how people find, save and share things on the web:

1. Capture as much as possible. Capture without judgment: your thoughts are more valuable than paper. Externalize what you learn. Once an idea is captured in a tangible form you can begin surveying and manipulating it.
2. Organize only after you capture. Filter, but don't delete irrelevant information. Computers are big enough to search and store everything. Make them manage it.
3. Synthesize into new meaning. Re-contextualize what you learn. This is the creative act. Experience becomes art, notes become a novel.

Most creative bottlenecks happen when you try to organize and synthesize before you capture. Spread it all out before you try to arrange it.

Many creative tools are designed for organizing first. Name the file before you can write in it. Create your Photoshop layer before you can draw on it. This causes friction.

Pascal's **Pensees** were collected mid-way through capture, organize, synthesize.

Design patterns

Design patterns are the solutions that have survived natural selection

Learn to learn

Learning to learn is a process of bootstrapping — getting better at getting better

---

Backlinks

Intuition is pattern re… | Gimmick Book | Create choices, mak…

Related

Questions are places… | OODA loop | Jack principles
Pace layers | The Art of Game Des… | Thing from the Future
Create choices, mak… | Brainstorming | Low floor, wide walls,…
Solvitur ambulando | Build to think | Learn to learn

Building Block Hypothesis

The building block hypothesis (Goldberg):

Genetic algorithms select for compact genes that encode impactful traits, because a short gene is less likely to be disrupted by mutation and crossover than is a long gene, which has more exposed surface area to disrupt.

These short-but-valuable genes become "building blocks" for evolution, allowing for composition at the level of traits — a **composable alphabet**.

Composable alphabets

Good alphabets compose at multiple levels

Goldberg proposes the Building Block Hypothesis for **Genetic Algorithms** in particular. Are there other places where it may apply? Stories, ideas, aphorisms?

Evolution

Evolution is a behavior that emerges in any system that has:
Mutation Heredity Selection
Evolutionary systems often generate unexpected solutions

Backlinks

Small alphabets

Related

Games with more th…
Second System Syn…
Evolution
Evolution is a diversif…
Capping downside di…
Containment can lea…
Punctuated equilibrium
TIMN
Slowly, then all at once
Ecology meta-patterns
Composable alphabets
Soft security

Questions are places in your mind where answers fit

> Questions are places in your mind where answers fit. If you haven't asked the question, the answer has nowhere to go. It hits your mind and bounces right off. You have to ask the question – you have to want to know – in order to open up the space for the answer to fit.

— Clay Christiansen

Also: **learn to learn**.

Related

Be approximately right
Intuition is pattern re…
The Feynman method
Pensees
Create choices, mak…
OODA loop
Low floor, wide walls,…
Brick pencil
Capture, organize, s…
Mindmap books
Solvitur ambulando
3 is the magic number

Modularize later

A principle from The Innovator's Dilemma. An innovation always starts out integrated, and is modularized later.

The boundaries of modularization matter. An object can be sliced in an infinite number of ways. What is important is that you slice it in a useful way.

Module boundaries should be drawn when a distinct sub-problem has been clearly identified in the value chain. At this point it can be farmed out.

By definition, an innovation is not well-understood before it has been created. This is why innovative products and technologies start off integrated, and why modularizing too early will fail. For a new product category, the value chain is just forming, the problem space is not well understood, and sub-problems are too ill-defined to be farmed out.

It is difficult to change module boundaries once drawn. After roads are laid down for a city, buildings may change, businesses may change, but roads rarely change.

Modularized products have a hard time adapting when the basis of competition changes, and the value chain is reorganized.

Modularizing along the wrong boundaries is the cause of **second system syndrome**.

Because innovations start off integrated, then become modularized, markets often have 2 phases:

- New market creation (integrated product)
- Modularization and commidification (modularized product)

Also: **Design patterns**, The Innovator's Dilemma, Disruption Theory.

Related

Prototyping	Job-to-be-done	Places to intervene i…
Lindy effect	4 ways to augment t…	Networked things
Macros are for when …	General-purpose lan…	Hash string UX
Slowly, then all at once	Start with a toy	Value, scale, time

Conformability

Looking through the lens of conformability means looking at what isn't changing, and asking how you can tailor your product/service/technology to conform to that landscape.

What changes slowly?

- Our **fundamental needs** — food, water, shelter, energy.
- People! Our physiology — hands, feet, head. There's a reason that ancient cuneiform tablets and iPhones have the same form factor.
- Infrastructure tends to change slowly. Roads, once laid down, don't often get moved.
- Cities change slowly. **Cities outlive civilizations**.
- Clothing. Conventions around pockets, bags, clothes tend to change on the decade timescale.

Thinking about **pace layers** helps identify the rate of change.

Related

Games with more th…
Genetic algorithms
Requisite Variety
Generalized problem…
Gimmick Book
Networked things
Goodhart's law
Slowly, then all at once
Punctuated equilibrium
100 Users
Scenario planning
Cities are like compo…

TUI

TUI — "tangible user interface" is the driving notion behind Hiroshi Ishii's research group at MIT Media Lab. The design goal of TUI is to make computation physical — something you can touch, smell, taste, hear.

musicBottles 2000 from Tangible Media Group on Vimeo.

> Music Bottles (1999) — information (music) is broken down into multiple channels, and each channel "bottled up". Uncorking a bottle releases its sound.

The shape of an object embues it with meaning. The shape of a tool can change the way we think.

> **Brick pencil**
>
> Interfaces matter

Ishii identifies 3 categories of TUI:

- Surfaces are for looking, and more recently, for touching. Because dislay technology is fairly advanced, we've spent the most time developing this mode of human-computer discourse.
- Tangibles: physical interfaces for digital interactions.
- Ambients change the mood of the space in living response to information. Ambients give you a sense of the shape of some computation, without demanding direct engagement. They might leverage color, sound, smell, to change the feel of a space. Over time, an ambient user interface might develop into an intuitive sensory extension. **Intuition is pattern recognition**.

Screens are extremely good at delivering programmable color. Shape and form are more difficult to program — but you can augment physical interactions with projection mapping, sound, haptics, to make objects feel more "charmed" than they really are.

Tangible CityScape from Tangible Media Group on Vimeo.

> Tangible Cityscape (2013) — a 3d program is made concrete, with physical pixels that actuate on the y-axis.

Where are the smarts? IoT says the smarts are in the thing, but Music Bottles and Dynamicland both take an alternative approach. Some smart centralized brain — a base, camera, or projector — can infuse ordinary objects with magic.

> Dynamicland uses dumb objects and a smart projector to make everything a little smart.

When approaching information and computation, ask:

- How can I make this physical?
- Can I engage with more than one sense at a time?
- Is there more than one way to interact with the object?
- No need to restrict yourself to one representation. Can I express different aspects of my computation as more than one concrete physicality?
- Can I imagine a tangible shape for a popular digital interaction?
- How do AR and VR change the TUI taxonomy?

Related: **4 ways to augment thought**.

Related

Invisible Cities	Thing from the Future	Gimmick Book
Create choices, mak…	A Pattern Language	Architects vs garden…
Desirability, feasibilit…	The Art of Game Des…	Jack principles
Conversational UI	Pace layers	Augmented convers…

Measure it

Two rules of thumb, seemingly in tension:

1. What's measured gets optimized
2. **Goodhart's law**: when a measure becomes a target, it ceases to be a good measure.

Just because you can measure it doesn't make it important. We tend to measure what is easy to measure.

We are trained to push back on assertions with a lack of evidence, but rarely question whether the available evidence is telling us the full story. If you look at the wrong thing, you'll arrive at the wrong conclusion.

> The data that is uncollected is infinite. What makes you think the data you do have is the right data? (Asymcar #36)

There is always bias in the data, because there is always a someone who collected the data and an instrument that allowed the colection to be done.

Horace Dediu:

> There is a hidden benefit to not having this data. All data is a creation and it tends to lead thinking in directions led by whatever is being measured (and whoever chose those

> measures and their motives). And yet without data there is no evidence and no credibility. In other words: You can't manage without measurement but you can't be sure what to measure.

Truncation (statistics).

Data can only measure the past. If the future looks like the past, then this can tell you the future. If the future doesn't look like the past... (see **Scenario Planning**)

The goal of creating a new product is to make the future look different from the past.

Exponent:

> The causal mechanism of being disrupted is looking at opportunities only in terms of numbers instead of looking at what causes the numbers.

> We should be guided by theory, not by numbers.
> — W. Edwards Deming

More: Capitalist's Dilemma, Who Solved the Capitalist's Dilemma?, A Capitalist's Dilemma (NYT), Toyota Production System, Beginner's Mind, Asymcar 10, Exponent 35.

Backlinks

Start with a toy

Related

Lindy effect
Bootstrapping
Games with more th…
Goodhart's law
Job-to-be-done
Write the press release
Start with a toy
Conformability
Amara's Law
Gall's Law
Gimmick Book
Generalized problem…

Charlie Kindel's 5 Ps

Charlie Kindel's 5 Ps for starting a project:

- Purpose: Why do we exist? Why are we in business? Where do we want to be in the future? What will we deliver?
- Principles: What are the non-negotiable rules and key strategies? How will we act?
- Priorities: What's the framework for tradeoffs?
- Plan: How are we going to stage and tackle solving the problems? What are the known dates & forcing functions on the calendar?
- People: Who's accountable for every key part of the plan?

Backlinks

Create choices, mak…
3 Ps
Project management…

Related

3 Ps
Soft security
Culture is a shared …
Start with a spreadsh…
Value, scale, time
Prototyping
Modularize later
Schein's model of or…
Start with a toy
100 Users
Project management…
Pournelle's Iron Law …

Second System Syndrome

Second System Syndrome is the curse by which a simple system is doomed to be replaced by an excessively abstract, over-engeered, or bloated successor.

You hack together a small simple program to solve a problem.

Congratulations! It's wildly successful. Now you have a community, and the limits of your hasty work are starting to show. Time for a rewrite!

You've been keeping a list of all the features you would like, limitations you would like to surpass, and you want to avoid getting boxed into a corner again, so you design a super abstract architecture.

Fast forward a few weeks, and you're tangled up in abstractions. The features you have built out don't add up to a product-market fit. You're still weeks away from a finished product. Hmm.

The phrase was first coined in the book The Mythical Man-Month.

How do you avoid it? **Start with a toy**, and **modularize later**.

See also: second system effect on Wikipedia, **Gall's Law**, **design patterns**.

Backlinks

How human systems…

Related

Low floor, wide walls,…
A Pattern Language
Verbs that can act on…
Places to intervene i…
Generative Grammar
Architects vs garden…
Gall's Law
Compositionality is c…
Slowly, then all at once
Prototyping
Language is a Lamar…
Goodhart's law

100 Users

> One of the most interesting ideas at Parc was: “every invention has to be engineered for 100 users”. So if you do a programming language or a DTP word processor, etc, it has to be documented for and usable by 100 people. If you make a personal computer, you have to be able to make 100 of them. If an Ethernet, it has to connect to 100 devices, etc.

— Alan Kay, What made Xerox PARC special?

PARC, MVP, **Design Patterns**

Related

Measure it
4 ways to augment t…
Slowly, then all at once
Prototyping
Value, scale, time
Start with a spreadsh…
Write the press release
Modularize later
Gall's Law
Games with more th…
Be approximately right
Places to intervene i…

Lindy effect

The Lindy Effect posits that the lifespan of an idea is about 2x as long as it is old. If the story of Beowulf has been told for about 1000 years, we can expect it to be told for 1000 more.

Unlike mortals and machines, ideas don't wear out with use. The more an idea is used, the likelier it is to stick around. What applies to ideas also applies to things powered by ideas — technologies, companies, cultures, religions. The general notion is that things which stand the test of time have properties which will allow them to continue to stand the test of time.

There is a kind of tautology here — by this rule, all ideas should be immortal. So what kills an idea? Ideas are like lobsters. They never die naturally. Eventually they get done in by some environmental factor.

Related: German tank problem and the Doomsday argument, **Pace layers**, **cities outlive civilizations**.

See Lindy Effect on Wikipedia.

Related

Gall's Law
Slowly, then all at once
Generalized problem…
Start with a toy
100 Users
Aurora
Be approximately right
Places to intervene i…
Modularize later

Measure it Manifesto Value, scale, time

LOCKSS

LOCKSS — Lots Of Copies Keep Stuff Safe.

> Evidence suggests that human factors (intentional and unintentional) are the greatest cause of loss or corruption to digital materials. Technology failures, economic failures and social failures all pose threats to the protection of digital content. When content is held in a single centralized repository, it is easy for people to tamper with a master copy without detection. A dark archive into which content disappears, only to reappear in a future emergency, does not engender confidence in either its availability or its correctness.

Backlinks

Top 1000 scientific p…

Related

Top 1000 scientific p… Papers are more dur… Top 1000 tools

General-purpose languages eat everything around them

Given time, a Turing-complete language will eat every domain-specific language adjacent to it.

Even on the web, where declarative, domain-specific languages had a 20 year head-start, JavaScript is eating HTML and CSS (React, CSS-in-JS).

Why? Turing-complete languages are universal machines:

- They can replace any domain-specific language.
- They transfer optionality from the platform implementation to 3rd party developers.

Once upon a time, there was a clever young woman who lived in a cottage by the sea, building an application platform.

Every day she would tend to the platform gently, adding features to a domain-specific language one by one, watching how the platform would respond and grow. Every day, she pruned the implementation, improving its performance.

She was thoughtful, so she knew that building a platform was **like growing a garden**. Markets could be trained, **but not planned**. Cultures could be grown, but not commanded into existence. She knew her platform was a complex adaptive system undergoing constant **evolution**, and filled with **agents** who would react to change. There were humans in the loop, so **Goodhart's law** ruled the day.

Life continued like this. Each day she cultivated her application platform, and each day it flourished a little more.

Then one day, she realized her ecosystem had grown so much that she could no longer keep track of the use-cases. The feature requests were greater than she could count! Click states, hover states, touch gestures, different controller types, reacting to user input... How would she fit all of these use-cases into her domain-specific language?

The following day a peddler came to her cottage, dressed in rags, and selling trinkets. Among the wares was nestled a small black box, the lid sealed with wax.

"Ah! This magic black box contains a Turing-complete scripting language. Open the box and all your ecosystem's problems will solve themselves. But I warn you — a Turing-complete spirit never travels alone. Once out of the box, it won't want to get back in."

The young woman, at her wit's end, took the Turing box and pried open the lid. Out rushed a Turing-complete spirit. Instantly, her ecosystem was transformed!

Wherever Turing-complete touched her domain-specific language, it was transformed into a 3rd party library! **Permissionless innovation** spread, and 3rd party libraries sprung up like weeds, creating features from nothing.

But chained to the tail of Turing-complete were 3 evil spirits — Performance, Privacy, and Security.

"Haha! Turing-complete may have transformed your ecosystem, but it is also under our curse! Wherever Turing-complete goes, we go. Whatever Turing-complete transforms, we spoil. You have transferred optionality from the platform to the scripting language and so you will be plagued by performance, privacy, and security problems forever!"

The young woman tried to force the spirits back into the box, but to no avail. Wherever Turing-complete went, the ecosystem flourished, but performance, privacy, and security problems sprang up too.

The next year the peddler came back. The young woman ran to him, saying "Turing-Complete has blessed my ecosystem, but everywhere it goes, it drags these 3 demons with it. Tell me peddler," said the young woman, "who can break their curse?"

"Ah young woman! I see you have opened the box. With all Turing-complete spirits come the blessings of **permissionless innovation**, and also these 3 curses. The curses cannot be broken. Once you transfer optionality from the platform to the scripting language, the deal cannot be undone. Never again will you go back to your domain-specific language.

"But all is not lost... East of the sun and West of the moon there are 3 mountains. In the heart of these mountains are 3 amulets: Sandbox, Neverslow, and Incentive. To obtain the amulets you must complete many trials. The amulets cannot break the curse, but they can tie the hands of the demons."

Over the years, the young woman went through many trials. She obtained the amulets, subdued Performance, Privacy, and Security — though never completely. Yet, her domain-specific language was gone forever, replaced by the magic of the black box, and when she returned home, she was a different person.

That is the way of things in real life. Fairy tales have a happy ending, but real life is full of incomplete solutions.

Advantages of declarative, domain-specific languages (DSLs):

- **USLICE**
- Sidesteps security, privacy, and performance problems.
- Implementation is decoupled from intent. The platform/OS has the option to optimize the performance of the implementation over time.
- Accessibility and other features can be baked into the language, rather than bolted on as a feature.

Problems with general-purpose languages (GPLs):

- Security, privacy, and performance problems. They can be mitigated with difficulty, but are at

some fundamental level, unresolvable.

- A Turing-complete script is a black box. There could literally be anything in there. The platform can't optimize, reimplement, or recontextualize scripted applications for other contexts (accessibility) without risk of breaking things.

The one massive advantage of a general purpose language: **permissionless innovation**. 3P developers can and will build anything they want.

Related

Slowly, then all at once	Not Rocket Science …	Markets do not make…
Small alphabets	Evolution is a diversif…	Provoking emergence
Goodhart's law	Containment can lea…	Places to intervene i…
Architects vs garden…	Modularize later	Genetic algorithms

Jack principles

The Jack Principles of **conversational UI** are design guidelines for the early text-based quiz game "You don't know Jack". There are some excellent design patterns here.

Maintain pacing:

1. Give the user only one task to accomplish at a time.
2. Limit the number of choices the user has at any one time.
3. Give the user only meaningful choices.
4. Make sure the user knows what to do at every moment.
5. Focus the user's attention on the task at hand.
6. Use the most efficient manner of user input.
7. Make the user aware that the program is waiting.
8. Pause, quit or move on without the user's response if it doesn't come soon enough.

Respond with human intelligence and emotion to:

1. The user's actions
2. The user's inactions
3. The user's past actions
4. A series of the user's actions
5. The actual time and space that the user is in

6. The comparison of different users' situations and actions

Maintain the illusion of awareness:

1. Use dialogue that conveys a sense of intimacy
2. Make sure characters act appropriately while the user is interacting
3. Make sure dialogue never seems to repeat
4. Be aware of the number of simultaneous users
5. Be aware of the gender of the users
6. Make sure the performance of dialogue is seamless
7. Avoid the presence of characters when user input cannot be evaluated

The original document (pdf): The Jack Principles

Also: **design patterns**, **conversational UI**.

Related

Recreate a piece of art
Thing from the Future
TUI
Augmented convers…
Human scale tools
Human scale
Gimmick Book
Capture, organize, s…
Pace layers
The Art of Game Des…
Understanding Comics
Architects vs garden…

Procedural paintbrush

Procedural paintbrushes are creative tools that augment your input.

An insight from Procedural Generation in Game Design, Generative Art Toys, by Kate Compton. A good procedural paintbrush is...

- Smarter than a pencil
- Constrains and magnifies your input
- Trades control for superpowers
- Defines a possibility space of mostly satisfying outcomes

Examples

- The pottery wheel constrains input to the x axis, and gives you symmetry superpowers in return.
- Spirograph constrains input to the choice of gears, but produces a variety of surprising outcomes.

Lenses

- When choosing constraints, consider **Gall's Law**.
- Should your procedural paintbrush create surprising results? Consider **architects vs gardeners**.
- Can this tool be used in **provoking emergence**? Where does user control begin

and end?
- Should feedback be direct, or indirect?

Backlinks

Brick pencil
Human scale tools

Related

Alphabets
Building Block Hypot…
Composable alphabets
Agent
Evolution
Generative Grammar
Small alphabets
Human scale tools
Ecology meta-patterns
Brick pencil
Low floor, wide walls,…
Tools over technique

Generative Grammar

Generative grammars let you build generative systems that produce variety, and structure.

A generative grammar describes a set of rules for constructing meaningful sentences from individual words. "Sentence" and "word" are meant loosely — really, grammars can be used to build many kinds of structures from smaller component parts. Textile designs, automatic poetry creators, and procedural level builders are all places where grammars could be used.

Grammars are DNA for your generator
DNA is a generative grammar with 4 "words" — GATC. **Evolution** is the process by which those words are expanded into a wide variety of "sentences", including you.

Taking a cue from nature, we can use generative grammars to construct "DNA" for generative programs:

- Generate a series of drawing commands using a grammar.
- Design game levels and rooms using a grammar.
- Describe cathedrals, mandalas, rose windows, city plans, or haikus using a grammar.

What does a grammar look like?

Grammars are often defined using named lists of templates:

<story>:
- Once upon a time, there was a <hero> who lived in a <location>...
- A long time ago, in a <location> far, far away, there was a <hero>...

Each template variable references another list of templates, which may reference other templates, and so on... describing a graph of relationships between templates.

<location>:
- <adjective> wood
- hut at the edge of a <landmark>
- castle far away

A template variable can be replaced with any item from the corresponding list of templates, and the resulting sentence will be valid. Structural correctness is encoded within the grammar itself.

By walking this graph of relationships, and making choices at each branch, we can expand the grammar into sentences:

Once upon a time, there was a kind old fisherman who lived in a hut at the edge of a quiet river.

The choices we make at each branch might be random, weighted, or determined by some other factor. The graph of relationships defines the structure, the choices produce variety.

Lenses

- When you see something that exhibits structure and variety, ask "can I describe this as a grammar?". Now, you have a recipe for generating that thing.
- Some grammars result in **evolution**. Others don't. What would it look like to introduce

mutation, heredity, or selection to your grammar?

Resources

- Generative Grammar on Wikipedia
- RiTA is a natural language toolkit and includes generative grammar helpers (JavaScript, Java).
- Tracery is a generative grammar library by Kate Compton (Javascript).
- Cheap Bots Done Quick lets you create generative Twitter bots using generative grammars.
- So you want to build a generator includes a section on grammars.

Backlinks

Agent

Related

Containment can lea…	Magic circle	Capping downside di…
Gall's Law	Language is a Lamar…	Permissionless creat…
Punctuated equilibrium	Verbs that can act on…	Markets do not make…
Composable alphabets	Conformability	Extrastatecraft

Write the press release

Amazon.com has a product development process they call Working Backwards. Rather than starting from an idea or a technology, you start by writing the press release for the product.

- The target audience is the customer.
- The press release is circulated internally.
- If the press release isn't compelling, you don't have a strong product hypothesis. Iterate! Iterating on a press release is less expensive than iterating on a product.

References:

- Working Backwards, Werner Vogels, CTO - Amazon.com
- Quora — What is Amazon's approach to product development and product management?

Related

Gall's Law
Job-to-be-done
Places to intervene i…
Lindy effect
Games with more th…
Manifesto
Measure it
Amara's Law
OODA loop
Tools over technique
Modularize later
Gimmick Book

Hash string UX

There are big UX differences between hash representations, and using the wrong hash representation can open the door to phishing scams.

TLDR: Special characters are snares. Base58 is great.

Make it easy to see at-a-glance
All hash strings are equally meaningless, but some are more confusing than others.

// Visually uniform, single shape
Z10KB012LOWEH30B09UIDK2L34E

// Visually chaotic, many shapes
A~o]O;%1hIi_-$1;:134).?=3aH
Rather than drawing attention to the individual characters, we want to make it easy to perceive the hash as a single unit (Gestalt principle). Having a more visually uniform string of characters helps us see the hash as a single shape, rather than a collection of individual shapes.

Make it easy to copy
Try double-clicking somewhere in the middle of this hash.

&Pe5kTo/V/w4MToasp1IuyMrMcCkQwDOdyzbyD5fy4ac=.sha256
You would expect to have the entire hash string selected on double-click, but the special

characters break it up into separate chunks.

Special characters don't just break up the hash visually, they also break it up programmatically, thwarting the text-selection affordances of most operating systems.

This forces users to manually select the text range, and increases the chances of mistakes. Multiply this error rate over every time a user has to select a hash — this causes serious friction.

Make it hard to spoof

Bitcoin goes one step further and uses Base58 for hashes, because it will not produce visually identical looking characters.

From base58.h in the Bitcoin source code:

```
/**
 * Why base-58 instead of standard base-64 encoding?
 * - Don't want 0OIl characters that look the same in some fonts and
 *      could be used to create visually identical looking data.
 * - A string with non-alphanumeric characters is not as easily accepted as input.
 * - E-mail usually won't line-break if there's no punctuation to break at.
 * - Double-clicking selects the whole string as one word if it's all alphanumeric.
 */
```

This prevents phishing-type attacks where the attacker spoofs an address using characters like O0, or Il1 to create identical-looking addresses.

Related

Judo move
Modularize later
Networked things
Pensees
Second System Syn…
USLICE
Not Rocket Science …
TIMN
General-purpose lan…
Prototyping
Soft security

Intuition is pattern recognition

Intuition is a type of pattern recognition. So:

1. Intuition can be trained by example.
2. Garbage in, garbage out.
3. Trust good taste.

> In solving a problem, the answer must be guessed at before a proof can even begin, and guesses are usually made from a knowledge of facts, experience, and hunches.
>
> The truly creative mathematician must be a good guesser first and a good prover afterward.
>
> -- Mathematics and Plausible Reasoning, Vol 1, George Polya

Taste is not "just a feeling". Taste is the result of your mind weighing hundreds of factors against prior experience and approximating a score.

Taste is powerful because it's fuzzy. **Be approximately right**.

Design patterns are patterns that have been recognized to work at social scale.

See also: Kill Math, **Capture, organize, synthesize**, **Ma**, **learn to learn**.

Backlinks

TUI

3 is the magic number

Related

4 ways to augment t…
3 is the magic number
OODA loop
Brick pencil
Create choices, mak…
Capture, organize, s…
Low floor, wide walls,…
Be approximately right
Solvitur ambulando
Learn to learn
The Feynman method
Questions are places…

Prototyping

Prototypes are something you build to convince yourself. Demos are something you build to convince others.

You build a prototype to answer a question. What is the question your prototype answers? When you define this question, it shows you what you should build and what you should fake.

When prototyping, think of the cheapest way to answer your question. There is an even cheaper way to answer your question.

In industrial design, there are "looks-like" prototypes and "works-like" prototypes.

- Looks-like: car chassis, painted, finished, without an engine.
- Works-like: engine without a car.

Building many small prototypes will always be easier than building one mega-prototype.

If you find yourself building One Prototype to Rule Them All, you are not building a prototype, you are building a product.

There is another kind of prototyping, which I call **build to think**. It is open-ended play with an eye toward discovery. When building to think, it's smart to define your timeframe and runway.

Every demo is built for an audience and a goal.

- Who is your audience?
- What do they want?
- What are they apprehensive about?
- How should the demo change their mind?

Backlinks

Solvitur ambulando
Create choices, mak…

Related

Start with a spreadsh…
Modularize later
Value, scale, time
100 Users
General-purpose lan…
Not Rocket Science …
Charlie Kindel's 5 Ps
Hash string UX
Judo move
Networked things
USLICE
Start with a toy

Genetic algorithms

Genetic algorithm **design patterns**:

- Principle of Meaningful Building Blocks. Learn from the **building block hypothesis** and find a coding that can express short, low-order schemata that are relevant to the underlying problem and relatively unrelated to schemata over other fixed positions.
- **Principle of Minimal Alphabets**: design the the smallest genetic alphabet that permits a natural expression of the problem.

From How genetic algorithms work.

> **Provoking emergence**
>
> So how do you provoke emergence? It's a process of gardening, not architecture

> **Evolution**
>
> Evolution is a behavior that emerges in any system that has:
> Mutation Heredity Selection
> Evolutionary systems often generate unexpected solutions

Related

Places to intervene i… Agent Requisite Variety

Second System Syn…
Ecology meta-patterns
Procedural paintbrush
Capping downside di…
Provoking emergence
Compositionality is c…
Goodhart's law
How human systems…
A Pattern Language

4 ways to augment thought

In Augmenting the Human Intellect, Doug Engelbart identifies 4 places where thinking can be augmented:

1. Artifacts — tools for manipulating materials and symbols.
2. Language — symbols we assign to concepts, allowing us to manipulate concepts at higher levels.
3. Methodology — procedures used in problem-solving.
4. Training — conditioning needed to practice 1, 2, and 3.

> The system we wish to improve can thus be visualized as comprising a trained human being, together with their artifacts, language, and methodology.
>
> — Augmenting the Human Intellect
> Doug Engelbart, 1962

Engelbart invented the mouse, and was the ringleader behind the Mother of All Demos. His lifework was dedicated to **bootstrapping** human thought, guided by the insight that our tools define how we may think — see **brick pencil**.

Backlinks

TUI

Human scale tools

Related

Gall's Law

Value, scale, time

Slowly, then all at once

Write the press release

Thing from the Future

Measure it

Be approximately right

Places to intervene i…

Goodhart's law

Capture, organize, s…

The Feynman method

Start with a toy

Human scale

All stories should happen at human scale, because meaning happens at human scale.

- Statistics w/o human stories attached fail to convey meaning.
- All numbers ending in "illion" are functionally equivalent for the purpose of pursuasion.

Design that ignores human scale won't be livable for humans.

- This is one of many critiques leveled at Modernism. People aren't platonic solids. See Brazil's dazzling modernist capital.
  - What I love about Frank Lloyd Wright and the Prarie School: the low, horizontal architecture exists at a human scale. It feels livable.
- Most of **A Pattern Language** is about designing cities for human scale.

> **Design patterns**
>
> Design patterns are the solutions that have survived natural selection

---

Backlinks

Zoom in, zoom out
Human scale tools
Cities are like compo…

Related

Once there was a TUI
Expectation, silence,…
Build to think
Desirability, feasibilit…
Aspect-to-aspect
We mistake stories f…
Augmented convers…
The Art of Game Des…
One surprise
Ma
Capture, organize, s…

Slowly, then all at once

How complex systems fail: slowly, then all at once.

All complex sytems are full of non-linear feedback loops. In a healthy complex system, these will have balancing feedback loops, holding each other in dynamic equilibrium. Given a small purturbence, these systems tend to fall back toward equilibrium, but a complex system is only ever metastable. If the purturbation is too intense, the feedback loops that make it up will fly apart, resulting in a sudden failure.

Pace layers: different systems seem to boom and bust on different timelines. Failures at lower layers tend to create catastrophic discontinuities for those above.

Related

Building Block Hypot…
Write the press release
Soft security
Composable alphabets
Bootstrapping
Language is a Lamar…
Extrastatecraft
Goodhart's law
Lindy effect
Generative Grammar
Places to intervene i…
Value, scale, time

Gall's Law

> A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.
>
> – John Gall (Systemantics: How Systems Really Work and How They Fail)

Evolution

Evolution is a behavior that emerges in any system that has:
Mutation Heredity Selection
Evolutionary systems often generate unexpected solutions

> Simple rules produce complex behavior.
> Complex rules produce stupid behavior.
>
> – Andrew Hunt, 37 Signals

See also:

Provoking emergence

So how do you provoke emergence? It's a process of gardening, not architecture

Design patterns

Design patterns are the solutions that have survived natural selection

Backlinks

How human systems…
Second System Syn…
Start with a toy
Procedural paintbrush

Related

Generalized problem…
Measure it
Write the press release
Requisite Variety
Composable alphabets
Conformability
Generative Grammar
Containment can lea…
Manifesto
How human systems…
Lindy effect
Alphabets

Games with more than one player

Every game with more than one player becomes a game about the interactions between those players.

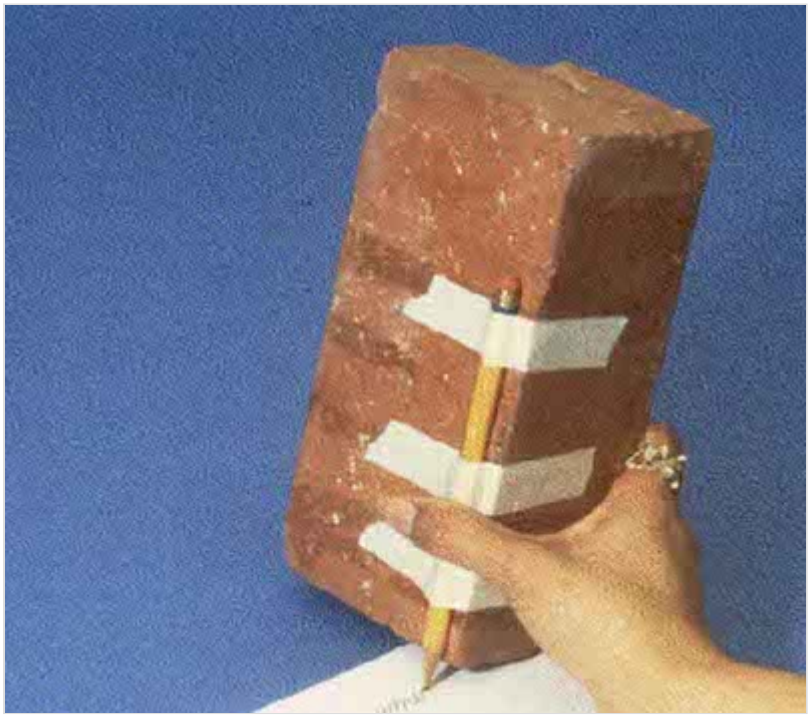
Also: **Provoking emergence**, **Villain, team, system**.

Related

Generalized problem…
Permissionless creat…
Composable alphabets
TIMN
Compositionality is c…
Soft security
Value, scale, time
The boundary of an …
Generative Grammar
Tools over technique
Agent
A Pattern Language

Brick pencil

Interfaces matter. Doug Engelbart taped a brick to a pencil to prove it.



Our interfaces influence the thoughts we think, and a bad interfaces interfere with thinking well.

From Augmenting Human Intellect: a Conceptual Framework (Douglas C. Engelbart, 1962):

> The Whorfian hypothesis states that "the world view of a culture is limited by the structure of the language which that culture uses." But there seems to be another factor to consider in the evolution of language and human reasoning ability. We offer the following hypothesis, which is related to the Whorfian hypothesis: Both the language used by a culture, and the capability for effective intellectual activity are directly affected during

> their evolution by the means by which individuals control the external manipulation of symbols. (For identification, we will refer to this as the Neo-Whorfian hypothesis.)
>
> Brains of power equal to ours could have evolved in an environment where the combination of artifact materials and muscle strengths were so scaled that the neatest scribing tool (equivalent to a pencil, possible had a shape and mass as manageable as a brick would be to us-assuming that our muscles were not specially conditioned to deal with it. We fastened a pencil to a brick and experimented. Figure 2 shows the results, compared with typewriting and ordinary pencil writing. With the brick pencil, we are slower and less precise. If we want to hurry the writing, we have to make it larger. Also, writing the passage twice with the brick-pencil tires the untrained hand and arm.

- Engelbart was trying to create better thinking tools, in pursuit of **bootstrapping**.
- Consider **tools over technique**.
- A **procedural paintbrush** is any tool that extends and augments your creative input.

Backlinks

4 ways to augment t… TUI

Related

Low floor, wide walls,… We mistake stories f… Gimmick Book

Be approximately right
Conversational UI
Judo move
Access to tools
Human scale
Capture, organize, s…
Desirability, feasibilit…
Invisible Cities
Build to think

Learn to learn

Learning to learn is a process of **bootstrapping** — getting better at getting better.

> Reading (learning) is the ultimate meta-skill and can be traded for anything else.
>
> — Naval's Laws

What first caused you to learn to learn?

Backlinks

Questions are places… | Capture, organize, s… | Generalized problem…
The Feynman method | Intuition is pattern re… | Be approximately right

---

Related

Questions are places… | Capture, organize, s… | The Feynman method
Pensees | 3 is the magic number | Solvitur ambulando
Be approximately right | Recreate a piece of art | Create choices, mak…
Mindmap books | OODA loop | 4 ways to augment t…

Human scale tools

Strive to build tools at **human scale**.

> If a system is to serve the creative spirit, it must be entirely comprehensible to a single individual.
>
> The point here is that the human potential manifests itself in individuals. To realize this potential, we must provide a medium that can be mastered by a single individual. Any barrier that exists between the user and some part of the system will eventually be a barrier to creative expression.
>
> — Alan Kay

By favoring **tools over technique**, we can unlock human potential that might have been missed due to lack of access. When building tools, strive for **low floor, wide walls, high ceiling**.

> **Low floor, wide walls, high ceiling**
>
> "Low floor, wide walls" is the mantra that guides the design of MIT's Scratch language

---

How can you make tools accessible to a single individual when the problem space is complex? Consider making a **procedural paintbrush**.

Tools are just one of **4 ways to augment thought**. How can human-scale tools be complemented by human-scale language, methodology, and training?

Goal: **access to tools**, human-scale tools, for a range of **fundamental needs**.

Tools have highest creative leverage when human scale. Stories have highest emotional impact when **human scale**.

Related

Recreate a piece of art
Brick pencil
Low floor, wide walls,…
Gimmick Book
Capture, organize, s…
Desirability, feasibilit…
Invisible Cities
Thing from the Future
Understanding Comics
4 ways to augment t…
Human scale
The Art of Game Des…

Invisible Cities

Invisible cities is a book about ways of seeing. It sits at a balance point between poetry, architecture, design, deconstruction, and magical realism.

Marco Polo narrates. The book is composed of 55 prose poems, each about a city he has visited. Each city belongs to a theme:

1. Cities & Memory
2. Cities & Desire
3. Cities & Signs
4. Thin Cities
5. Trading Cities
6. Cities & Eyes
7. Cities & Names
8. Cities & the Dead
9. Cities & the Sky
10. Continuous Cities
11. Hidden Cities

These themes repeat in a sequences that resemble generative music. The structure is there, but it's hard to put a finger on what exactly the structure is.

The poems themeselves have recurring tropes that suggest an elusive pattern.

- Evocative materials: glass, bronze, bone

- Imports and exports
- Key locations: temples, markets, stables, towers

Each poem can be read as magical realism, a metaphor, or a way of seeing. For example:

> 4. Trading Cities
>
> In Ersilia, to establish the relationships that sustain the city's life, the inhabitants stretch strings from the corners of the houses, white or black or gray or black-and-white according to whether they mark a relationship of blood, of trade, authority, agency. When the strings become so numerous that you can no longer pass among them, the inhabitants leave: the houses are dismantled; only the strings and their supports remain.

Is this a description of a real city, or a description of all real cities?

Related

TUI
Conversational UI
Build to think
Human scale
Marchetti's constant
Gimmick Book
Thing from the Future
Recreate a piece of art
Augmented convers…
Architects vs garden…
Harvesting the Biosp…
Jack principles

Provoking emergence

So how do you provoke emergence? It's a process of **gardening, not architecture**. Collecting patterns...

Examples of emergent behavior

- Biological **evolution**
- Flocking
- Conway's Game of Life
- Wikipedia
- Etherpad + pirate party
- Minecraft
- urls and linking

Markets are an emergent behavior, so designing for emergence is how you create new markets.

Emergence design patterns

You can't top-down plan emergent systems, but you can provoke them into being. Here are some attributes that emergent systems often have:

- **Evolution**
- More than one goal, or constraints with no goal
- Goals that can be achieved in more than one way
- **Alphabets** - a kit of parts, together with rules for combining them.
- Universal APIs
  - Example: Lego dots.

    - Universal APIs allow anything to be combined with anything.
- **Verbs that can act on many objects**.
- Side effects that change constraints create a dancing landscape.
- **Fun to play with**.
    - This gets you past the steep part of the learning curve and into creating.
- **Low floor, wide walls, high ceiling** — an easy initial learning curve.
    - What can I do immediately? In Minecraft I can dig.
- Infinite spaces (or simulated by generative algorithms)
    - Examples: No Man's Sky, Spelunky.
- Involves more than one person
    - See **games with more than one player**.
    - Often has some network effect
    - Multipolar arenas (2 or more actors) produce complex outcomes that are difficult to anticipate.
    - Create campfires: make opportunities for subcultures to emerge.

Why provoke emergence? Because **permissionless creation is where value comes from**.

Design patterns

Design patterns are the solutions that have survived natural selection

Backlinks

Gall's Law
Genetic algorithms
Small alphabets
Containment can lea…
Composable alphabets
Procedural paintbrush
General-purpose lan…
Ecology meta-patterns
Compositionality is c…

Related

General-purpose lan…
Evolution is a diversif…
Building Block Hypot…
How human systems…
Compositionality is c…
Composable alphabets
Soft security
Alphabets
Goodhart's law
Requisite Variety
Genetic algorithms
Gall's Law

The Feynman method

Richard Feynman was a polymath, who did pioneering work in the Manhatten Project, quantum physics, parallel computing, and nanotechnology. One would expect he knows a thing or two about learning. Lucky for us, he boiled his technique down into 4 steps.

The Feynman method of learning:

1. Find a concept
2. On a piece of paper, write an explanation of the concept using simple words — the way you would teach it to a first-grader.
3. When you encounter a gap, or find yourself resorting to big words and tricky concepts, go back to the source material. Re-learn, until you can explain it in simple terms.
4. Share it!

Also: **learn to learn**, **be approximately right**.

Backlinks

Mindmap books

Related

Low floor, wide walls,…
Pensees
OODA loop
Capture, organize, s…
Intuition is pattern re…
3 is the magic number
4 ways to augment t…
Solvitur ambulando
Be approximately right
We mistake stories f…
Create choices, mak…
Learn to learn

Alphabets

Emergence springs from alphabets. An alphabet is a kit of parts, together with rules for combining them.

Christopher Alexander. "Systems Generating Systems" Architectural Design 38, no. December 1968 (1968): 605-610.

If you want to **provoke emergence**, design an alphabet.

Examples:

- The english alphabet has 26 letters, together with phonetic rules for combining them into words.
- Minecraft allows you to craft new items with new properties, using a kit of existing items, plus rules for combining them.
- Chemistry defines how you may combine molecules to produce new materials with properties different from the sum of the parts.
- DNA has a tiny alphabet ATCG, but the meanings this alphabet can produce have not yet been exhausted.

Composable alphabets

Good alphabets compose at multiple levels

Small alphabets

When designing alphabets for emergence, constrain the alphabet, but don't constrain what may be written with it

> Most designers today think of themselves as the designers of objects. If we follow the argument presented here, we reach a very different conclusion. To make objects with complex holistic properties, it is necessary to invent generating systems which will generate objects with the required holistic properties. The designer becomes a designer of generating systems — each capable of generating many objects — rather than a designer of individual objects. — Christopher Alexander, Systems Generating Systems, 1968

Backlinks

Verbs that can act on…
Language is a Lamar…

Related

Genetic algorithms
Containment can lea…
Capping downside di…
Punctuated equilibrium
Small alphabets
Compositionality is c…
A Pattern Language
Low floor, wide walls,…
Verbs that can act on…
How human systems…
Procedural paintbrush
Soft security

Small alphabets

When designing **alphabets** for emergence, constrain the alphabet, but don't constrain what may be written with it.

Constrain the alphabet. Keeping the alphabet small makes it easy to learn. It also forces you, the designer, to create a **composable alphabet**. With a smaller set of letters, expressiveness is achieved by designing rules that allow letters to be composed with each other in many different ways.

> **Composable alphabets**
>
> Good alphabets compose at multiple levels

...But don't constrain what may be written with it. **Alphabets** that produce open-ended emergence — written language, chemistry, DNA — don't place a limit on the number of **letters you may combine in sequence**, or the meanings you may construct with those letters. Anything achievable within the rules of the alphabet is permissable. This generates an open-ended possibility space.

- DNA has a tiny alphabet (ATCG), but no limit on the number of basepairs that may be chained together, and no constraint, outside of natural selection, on the phenotypes that may be generated.
- The web defines a set of technologies (HTML, CSS, JavaScript), and rules for combining

them, and does not censor what may be created with those technologies. This **permissionless system** continues to generate surprising new business models.

When working with **genetic algorithms** you want to define the smallest alphabet of genes that can pragmatically express the desirable morphospace.

> **Genetic algorithms**
>
> Genetic algorithm design patterns: Principle of Meaningful Building Blocks

> **Building Block Hypothesis**
>
> The building block hypothesis (Goldberg): Genetic algorithms select for compact genes that encode impactful traits, because a short gene is less likely to be disrupted by mutation and crossover than is a long gene, which has more exposed surface area to disrupt

---

Can we analyze App Stores through this same heuristic? Native apps have a large alphabet, in the sense of access to many low-level device properties. App review limits what may be written with this alphabet. This constrains **permissionless innovation** to those innnovations which are compatible with the host company's goals, rather than constraining innovation to that which is possible according to the rules of the alphabet.

Provoking emergence

So how do you provoke emergence? It's a process of gardening, not architecture

Backlinks

Verbs that can act on…

Related

Alphabets
Compositionality is c…
Evolution
Verbs that can act on…
Soft security
Evolution is a diversif…
Permissionless creat…
Slowly, then all at once
Extrastatecraft
Agent
The boundary of an …
Generative Grammar

Composable alphabets

Good **alphabets** compose at multiple levels.

- Letters combine into words, which describe ideas.
- DNA basepairs combin into genes, which encode traits.
- Minecraft lets you craft new blocks with new properties from combinations of other blocks.

When components of an alphabet are combined, they often express meanings which are different in kind to the alphabet that makes them up. Each new level of meaning forms another alphabet, at another level.

The most powerful alphabets enable composition at multiple levels. This allows for great expressive power across different kinds of meanings, at multiple levels of complexity.

Provoking emergence

So how do you provoke emergence? It's a process of gardening, not architecture

Backlinks

Building Block Hypot…
Verbs that can act on…
Small alphabets

Related

Architects vs garden…
Verbs that can act on…
Permissionless creat…
Evolution
Extrastatecraft
A Pattern Language
Pace layers
Low floor, wide walls,…
Evolution is a diversif…
Requisite Variety
Provoking emergence
Games with more th…

Brainstorming

The design process in a nutshell: **create choices**, **make choices**. Brainstorming is what you do to create choices.

Since the goal is to produce lots of raw material for further thinking and refinement, you should avoid critical modes of thinking at this stage. The goal is to go for quantity of ideas over quality of ideas.

Conducting Group Brainstorms
Conducting brainstorms in groups is nearly always better than going it alone. You don't know what you don't know, making it hard to explore a space all by yourself. A diversity of perspecives will create syntheses that no one person could have created alone.

Two key challenges to brainstorming in groups are setting up the ground rules so the brainstorm doesn't devolve into a critiquing session, and coming to consensus.

One common approch to solving this problem is the sticky note brainstorm.

Materials

- Sticky notes (colorful)
- Sharpies (medium felt tip)
- PaperMate Flair M (optional) for finer illustrations

Group homework before you start

Describe the user needs you're solving for, usually in the form of short user stories. These can be uncovered through user research, market research, user interviews, exploration or some other process. Ask "what is the **job-to-be-done**?" Its ok if you don't know for sure. In new market exploration, everything is a hypothesis.

Explore the problem space.

- What analogous problems are there in other fields?
- What other solutions are there? What isn't working?
- Look for users that are extreme outliers. What does their behavior tell you about the underlying **job-to-be-done**?

Conducting the brainstorming session

1. Set the ground rules and goals
   - Quantity over quality
   - No critiques
   - Build on ideas. Replace "yes, but..." with "yes, and..."
2. Distribute sticky notes, sharpies
3. Pose a question, user need or topic.
4. Participants have 5 min to get as many ideas on sticky notes as possible. 1 idea per sticky note. Sharpies help because they are nice and thick — legible at a distance, and they prevent fine detail work.
5. At the end of the 5m mark, go around and have people share their ideas.
6. Put the ideas up on the board.
7. Do this for 3 rounds. Toward the later rounds, new ideas tend to emerge, synthesized from

others already shared.

8. Cluster ideas by theme, topic, approach.

Coming to consensus: An optional step for coming to consensus is to allow everyone to signal their interest by voting. Each person gets 5 votes that they can allocate in any way, by taking a sharpie and adding a dot to the corner of a sticky note. This gives everyone a voice, and helps channel discussions about next steps in a meaningful direction.

Related

Create choices, mak…	Augmented convers…	Judo move
Pace layers	Invisible Cities	Jack principles
Capture, organize, s…	TUI	A Pattern Language
Recreate a piece of art	Human scale tools	Desirability, feasibilit…

Start with a toy

When designing something new, start with a toy.

- Toys are fun. Starting with one forces you to create something desirable from the start (see **Desirability, Feasibility, Viability**).
- Toys invite people to play. Play is how we learn new things.
- Toys don't have directed goals. When something is truly new, you don't know what it is good for. Use-cases don't exist yet. A toy lets you find your use-case through exploration.
- Toys don't threaten incumbents. Instead of squashing you, an incumbent will ignore you. (First they ignore you, then they laugh at you, then they fight you, then you win.)

Power users find it difficult to accept toys. They are used to the way the axe handle fits in the palm. The tool has become an extension of self. GUIs will never be as powerful as command lines. Laptops will never be as powerful as desktops. Phones will never be as powerful as laptops...

Large organizations find it difficult to justify funding toys. A large organization is hungry for large profit, but a toy can't justify itself with a large bottom line. This pushes incumbents toward well-trodden paths with measurable bottom lines and away from exploration.

> If it works, it's obsolete.

— Marshall Mcluhan

"Measurable" means someone else got there first and **measured it**. "User segment" means someone else got there first and segmented it.

If something is new, it has yet to be measured. It has no users. It also has no competition.

Start with a toy.

See also: **Gall's Law**, **Design patterns**.

Backlinks

Judo move
Containment can lea…
Provoking emergence
Second System Syn…

Related

Goodhart's law
Bootstrapping
Amara's Law
Games with more th…
Be approximately right
Places to intervene i…
Start with a spreadsh…
Manifesto
Conformability
Tools over technique
Charlie Kindel's 5 Ps
OODA loop

Low floor, wide walls, high ceiling

"Low floor, wide walls" is the mantra that guides the design of MIT's Scratch language.

> When discussing technologies to support learning and education, my mentor Seymour Papert often emphasized the importance of “low floors” and “high ceilings.” For a technology to be effective, he said, it should provide easy ways for novices to get started (low floor) but also ways for them to work on increasingly sophisticated projects over time (high ceiling).
>
> For a more complete picture, we need to add an extra dimension: wide walls. It’s not enough to provide a single path from low floor to high ceiling; we need to provide wide walls so that kids can explore multiple pathways from floor to ceiling.

— Mitchel Resnick

This gives us a lens through which evaluate tools:

- Low floor: how easy is it to learn?
- Wide walls: how inclusive is it? How many different use-cases does it serve?
- High ceiling: does it scale with growth?

Most programming tools focus exclusively on creating a high ceiling. By choosing the other two neglected dimensions, Scratch has carved out a durable niche for itself. It is widely used in intro-to-programming courses, beginner robotics, after-school classes... any place a simple but open-ended programming model is needed.

Product development has a natural bias toward pushing the ceiling higher — focusing on the needs of the most valuable customer, where the highest amount of value may be extracted. This is one of the driving forces of disruption in the Innovator's Dilemma.

How can your product focus on:

- Lowering the floor — becoming simpler, cheaper, "good enough"
- Widening the walls — serving an ignored audience or market

Design patterns, Toy

Backlinks

Human scale tools Provoking emergence

Related

TIMN
Genetic algorithms
Thing from the Future
Generative Grammar
Recreate a piece of art
Cities are like compo…
Gall's Law
Intuition is pattern re…
Architects vs garden…
Compositionality is c…
3 is the magic number
Extrastatecraft

We mistake stories for theories

We often mistake stories for theories. Stories are a series of causally linked events, but they aren't a rigorous theory of causation.

This conflation is heavily exploited by consultants.

See also: **storytelling**.

Related

The Feynman method
Capture, organize, s…
OODA loop
Questions are places…
3 is the magic number
Mindmap books
Three-act story struc…
In medias res
Pensees
Learn to learn
Meaning, detail, drama
Create choices, mak…

Evolution

Evolution is a behavior that emerges in any system that has:

1. Mutation
2. Heredity
3. Selection

Evolutionary systems often generate unexpected solutions. Nature selects for good enough.

> There is no such thing as advantageous in a general sense. There is only advantageous for the circumstances you're living in. — Olivia Judson, Santa Fe Institute

---

Evolving systems exist in punctuated equilibrium.

Punctuated equilibrium

Complex systems exist in punctuated equilibrium, repeatedly evolving through distinct phases of randomness, growth, consolidation, and collapse

---

Questions:

- What systems (beside biology) exhibit evolutionary behavior? Remember, evolution

happens in any system with mutation, heredity, selection.

- What happens to an evolutionary system when you remove mutation? Heredity? Selection?
- Do you see a system with one of these properties? How can you introduce the other two?

Provoking emergence

So how do you provoke emergence? It's a process of gardening, not architecture

Resources: Evolutionary Systems on Wikipedia.

Backlinks

How human systems…	Gall's Law	Containment can lea…
Requisite Variety	Genetic algorithms	Cities are like compo…
Ecology meta-patterns	Evolution is a diversif…	Generative Grammar
Agent	Capping downside di…	Building Block Hypot…
TIMN	The boundary of an …	Design patterns
Goodhart's law	General-purpose lan…	Language is a Lamar…

Related

TIMN	Genetic algorithms	Pace layers
Requisite Variety	Places to intervene i…	Conformability
Composable alphabets	Verbs that can act on…	Capping downside di…
Slowly, then all at once	A Pattern Language	Generative Grammar

Agent

From Signals and Boundaries, MIT Press: when a system is capable of reprogramming itself in response to outside signals, it can be modeled as an adaptive agent.

- Agents are enclosed within a boundary.
- Agents filter outside signals through boundaries.
- Agents are turing-complete. They may accept some signals and ignore others (if/else). Signals may be recursive (feedback loops). Boundaries are places where a feedback loop can be halted. They provide a "breakpoint" in the loop.
- Within the boundary, agents carry histories, or "state" — the result of their individual reactions to signals over time.
- Multiple agents can form conglomerates — higher levels of organization. Conglomerates can also be seen as agents.
- Agents reprogram themselves in response to outside signals - this means both adapting behavior to signals (if/else), and changing the structure of the "programs" that generate the behavior in the first place.

In machine learning, fitness pressure for **evolution** is often provided by a fitness function that produces a reward for agents that move closer toward some desired goal.

In a complex adaptive system, this doesn't cut it. The behavior of a CAS is too complex to be easily expressed as a fitness function. In a real sense, complex adaptive systems don't have a singular "goal". They are autotelic. To model these systems in a way that will produce realistic **evolution**:

- Agents are situated within a landscape. This landscape contains resources (e.g. food). Resources are heterogenous and distributed unevenly.
- Agents reproduce by collecting resources.

A challenge for modeling complex adaptive systems: finding a **generative grammar** that can produce both the signals and the boundaries of the system.

The actor model expresses computer programs as networks of adaptive agents.

Agent-based modeling is often used to derive models for aspects of complex adaptive sytems.

> Most can agents turn out be persistent patterns imposed on flows. As was mentioned earlier, the human body turns over most resident atoms within days, and no atom resides in the body for more than a year or so. Similarly, in a great city, individuals arrive and depart daily but the overall pattern of activity persists. — John Holland, Signals and Boundaries

Backlinks

Requisite Variety
The boundary of an …
Punctuated equilibrium
General-purpose lan…
Ecology meta-patterns
Goodhart's law

Related

Punctuated equilibrium
Low floor, wide walls,…
Pace layers
Permissionless creat…
Ecology meta-patterns
Procedural paintbrush
General-purpose lan…
Extrastatecraft
Small alphabets
Requisite Variety
Cities are like compo…
Places to intervene i…

Pensees

There are rumors Pascal wrote the Pensées on notecards, and pinned these cards to a wall, connecting related thoughts with yarn. An early example of hypertext?

> ...the Pensées, a pile of papers concerning religion, were originally written on mostly large sheets of paper, some of which were subsequently cut into individual passages, of which again only some were divided into twenty-seven bundles or laisse, 414 in all, or just under half of the total were then attached together by thread running through holes pierced in the top left corner and knotted after the title had been given to the group.

Pascal's Pensées were collected mid-way through **capture, organize, synthesize**.

Pascal's Pensées point toward an under-explored writing format — small, networked, loosely-connected passages, clustered by theme, with no single linear narrative.

This seems like a natural fit for hypertext.

Backlinks

Gimmick Book

Related

4 ways to augment t…	Create choices, mak…	Questions are places…
3 is the magic number	Mindmap books	Brick pencil
TIMN	Low floor, wide walls,…	Soft security
Recreate a piece of art	Be approximately right	Intuition is pattern re…

Desirability, feasibility, viability

IDEO famously framed successful products as having these ingredients:

1. Desirability: something people want
2. Feasibility: something that is possible
3. Viability: something that makes money

More:

- About IDEO
- IDEO and design thinking on Fast Company

Design patterns

Design patterns are the solutions that have survived natural selection

Backlinks

Start with a toy

Related

Recreate a piece of art
Create choices, mak…

A Pattern Language
Gimmick Book

Understanding Comics
Capture, organize, s…

Augmented convers…
Architects vs garden…
Brainstorming
Brick pencil
Build to think
Jack principles

Design patterns

Design patterns are the solutions that have survived natural selection. Somewhere between folklore and formalism, they point out the pivot points on problems.

Design patterns are unreasonably effective. They give you leverage even if you don't fully understand the problem space. The patterns that work, work, even when we don't know why they work.

The concept of "pattern" was defined in Christopher Alexander's **A Pattern Language**.

> **A Pattern Language**
>
> In A Pattern Language, Alexander argues for the primacy of design patterns as a language for creation, against top-down theory

See also:

> **Culture is a shared mechanism for problem solving**
>
> Edger Shine, on the culture of a business: Culture is a shared mechanism for problem solving

> **Evolution**

Evolution is a behavior that emerges in any system that has:
Mutation Heredity Selection
Evolutionary systems often generate unexpected solutions

Backlinks

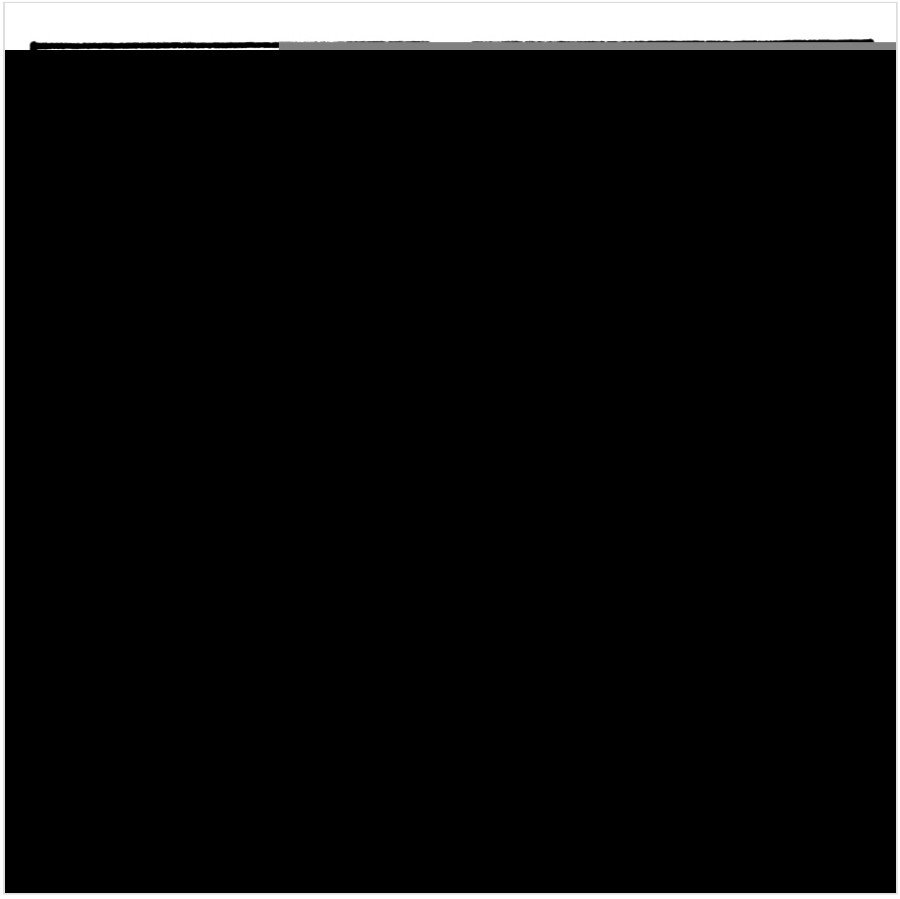
Human scale
100 Users
Marchetti's constant
One surprise
Provoking emergence
Second System Syn…
Gall's Law
Requisite Variety
Desirability, feasibilit…
Generalized problem…
Capture, organize, s…
Modularize later
Gimmick Book
Start with a toy
Genetic algorithms
Intuition is pattern re…
Low floor, wide walls,…
Jack principles

Related

Manifesto
Goodhart's law
Measure it
4 ways to augment t…
Conformability
Value, scale, time
Write the press release
Gall's Law
Gimmick Book
Slowly, then all at once
Amara's Law
Be approximately right

Understanding Comics

Scott McCloud's Understanding Comics introduces a vocabulary for visual storytelling. This vocabulary is used for comics, but also storyboarding, advertisements, movies — any medium where sequence and image are used to form narrative. The format of "comic book about comic books" allows him to to show, rather than tell how these techniques work.



Backlinks

Panel-to-panel

Related

Pace layers
Gimmick Book
Create choices, mak…
Architects vs garden…
Conversational UI
Jack principles
Build to think
Desirability, feasibilit…
Brainstorming
Invisible Cities
Augmented convers…
Thing from the Future

Augmented conversations

Something fascinating is going on with WeChat:

> WeChat has popularized the concept of “official accounts” for brands and public figures. They’re kind of like the IRC and AIM bots of yore — think SmarterChild but for banks, phone companies, blogs, hospitals, malls, and government agencies. Many institutions that otherwise would have native apps or mobile sites have opted instead for official accounts.
>
> You can send any kind of message (text, image, voice, etc), and they’ll reply, either in an automated fashion or by routing it to a human somewhere. The interface is exactly the same as for chatting with your friends, save for one difference: it has menus at the bottom with shortcuts to the main features of the account (though it can be toggled away to reveal the normal text field).

Because you're having a conversation over text, you can swap out the AI with a person on the other side. Conversational systems could be hybrid Augmented Intelligences

This approach is a bit like being able to cheat the Turing Test... passing a message from a human when convenient.

Also: **conversational UI**.

Related

A Pattern Language
Gimmick Book
Desirability, feasibilit…
Pace layers
Create choices, mak…
Judo move
Understanding Comics
TUI
Brainstorming
Human scale
Jack principles
Invisible Cities

Verbs that can act on many objects

When trying to **provoke emergence**, design verbs that can act on many objects.

From **The Art of Game Design**, by Jesse Schell:

> Verbs that can act on many objects. This is possibly the single most powerful thing you can do to make an interesting game. If you give a player a gun that can only shoot bad guys, you have a very simple game. But if that same gun can be used to shoot a lock off a door, break a window, hunt for food, pop a car tire, or write a message on the wall, now you start to enter a world of many possibilities.

By increasing the number of ways you may combine verbs and objects, you create an interaction **alphabet**.

It is possible to increase the composability of your alphabet even further, by using parts of speech.

From Controls as Language:

> It's always surprised me that most games don't really have any equivalent to adverbs: discrete ways of doing things, such as "quietly" or "aggressively", that can be applied to any verb in order to modify its effects.

> It's much easier to learn four generic verbs and three generic adverbs than twelve entirely separate verbs. Transfer of learning plays a major role here: if a player learns the adverb "quickly" in conjunction with the verb "walk", and then learns the verb "hit", they may be able to transfer their understanding of the generic adverb to the new verb and correctly intuit (without explicit instruction) that the combined phrase "hit quickly" may be used to make a rapid strike.

The typical user interface can be seen as an alphabet of verbs that compose with objects, but not with each other. A **smaller alphabet**, with a greater degree of composability can be easier to learn, and more expressive.

Another example, again, from Controls as Language:

> The Vim text editor makes use of an input language with distinct parts of speech rather than a conventional one-keyboard-shortcut-per-action editor control scheme, and is frequently lauded for its power and flexibility as a result.

When you make **small**, **composable** alphabets it **provokes complex emergence**.

- Where might we increase the number of objects a verb can act upon?
- How might we create **small alphabets** of actions, and increase the number of interactions?
- What side-effects do verbs have?

Related

Generative Grammar	Evolution	Second System Syn…
Pace layers	The boundary of an …	Compositionality is c…
Punctuated equilibrium	Agent	Gall's Law
Composable alphabets	Slowly, then all at once	How human systems…

The book will kill the edifice

> Architecture is the great book of humanity, the principal expression of man in his different stages of development, either as a force or as an intelligence.
>
> When the mass of reminiscences of the human race became so heavy and so confused that speech naked and flying, ran the risk of losing them on the way, men transcribed them on the soil in a manner which was at once the most visible, most durable, and most natural. They sealed each tradition beneath a monument.
>
> The first monuments were simple masses of rock, “which the iron had not touched,” as Moses says. Architecture began like all writing. It was first an alphabet. Men planted a stone upright, it was a letter, and each letter was a hieroglyph, and upon each hieroglyph rested a group of ideas...
>
> In the fifteenth century everything changes.
>
> Human thought discovers a mode of perpetuating itself, not only more durable and more resisting than architecture, but still more simple and easy. Architecture is dethroned. Gutenberg’s letters of lead are about to supersede Orpheus’s letters of stone.

The book is about to kill the edifice.

The invention of printing is the greatest event in history. It is the mother of revolution. It is the mode of expression of humanity which is totally renewed; it is human thought stripping off one form and donning another; it is the complete and definitive change of skin of that symbolical serpent which since the days of Adam has represented intelligence.

In its printed form, thought is more imperishable than ever; it is volatile, irresistible, indestructible. It is mingled with the air. In the days of architecture it made a mountain of itself, and took powerful possession of a century and a place. Now it converts itself into a flock of birds, scatters itself to the four winds, and occupies all points of air and space at once.

— Victor Hugo, Notre-Dame de Paris

Related

Aurora
Invisible Cities
Marchetti's constant
Scenario planning
Clark's 3 laws
Conformability
Amara's Law
Science is the only n…
Extrastatecraft
Thing from the Future
Lindy effect
A Pattern Language

Generalized problem solving

Stanley Kubrick: "develop a generalized method of solving problems".

> I think that if you get involved in any kind of problem-solving in depth, on almost anything, it's surprisingly similar to problem-solving on anything. I started out by getting a camera and learning how to take pictures, and learning how to print pictures, and learning how to build a dark room, and learning how to do all the technical things, and so on and so on. And then finally trying to find out how you could sell pictures and - would it be possible to be a professional photographer. And it was a case of over a period of say, from the age of 13 to 17, you might say going through, step by step by myself, without anybody really helping me, the problem-solving of becoming a photographer. And i found that, i think looking back, that this particular thing about problem solving is something that schools generally don't teach, and that if you can develop a generalized approach to problem solving that it's surprising how it helps you in anything.

From Stanley Kubrick Interview with Jeremy Bernstein, 27th November 1966

What is my personal generalized method of problem solving?

- Collect a Swiss-army knife of **design patterns**
- **Create choices, make choices**

Also: **learn to learn**

Related

Modularize later
Job-to-be-done
OODA loop
Amara's Law
Conformability
Bootstrapping
Write the press release
Goodhart's law
Gall's Law
Gimmick Book
Games with more th…
Lindy effect

Bootstrapping

Bootstrapping is the process of getting better at getting better. Things that bootstrap exhibit exponential improvement.

Doug Engelbart coined the term bootstrapping to describe the strategy at his Augmentation Research Center, of focusing on building tools for thinking powerfully, and building tools for making tools.

On paper, there are clear advantages to building tools to build better tools. One practical drawback is that bootstrapping has high up-front time-costs. Rather than focusing on finding a product-market fit, you're going one level deeper, one level deeper, one level deeper, to build the tool that builds the tool that builds the product.

This approach can work well in a research lab setting, where the runway is longer and the goals are not commercial profit.

Dilemma: can this approach work in a commercial setting?

References: Bootstrapping on dougenglebart.com

Also: **learn to learn**, **build to think**, **access to tools**.

Backlinks

4 ways to augment t…
Brick pencil
Tools over technique

Related

Games with more th…
Be approximately right
Lindy effect
Gimmick Book
Modularize later
Places to intervene i…
100 Users
Job-to-be-done
Write the press release
Manifesto
4 ways to augment t…
Generalized problem…

Job-to-be-done

Job-to-be-done theory is part of Clayton Christensen's thinking framework for investigating Disruption Theory. Rather than thinking in product categories and market segments, ask what is the job the customer is hiring this product to do?

A classic illustration of job-to-be-done thinking is Theodore Levitt's quip that people don't want to buy a quarter-inch drill, they want a quarter-inch hole.

If you're a drill company, missing this distinction causes you to fixate on competing symmetrically with other drill manufacturers on specs — drill speed, power, cost, whatever. Meanwhile some other easier way of making quarter-inch holes can come along and blow you out of the water.

Resources:

- Clay Christensen's Milkshake Marketing on HBS
- Understanding the Job - Clay Christensen (YouTube)
- Jobs-to-be-done — Clay Christensen (YouTube)

Related: The Inovator's Dilemma.

Backlinks

Brainstorming
Project management…

Related

100 Users
Amara's Law
Measure it
Be approximately right
Generalized problem…
Manifesto
Start with a toy
Tools over technique
Slowly, then all at once
Modularize later
OODA loop
Gimmick Book

USLICE

USLICE — properties that make the web powerful:

- Updatable
- Secure
- Linkable
- Indexable
- Composable
- Ephemeral

Backlinks

General-purpose lan…

Related

Judo move
Modularize later
Hash string UX
Second System Syn…
Prototyping
Not Rocket Science …
General-purpose lan…
Networked things

Tools over technique

> The great benefit of computer sequencers is that they remove the issue of skill, and replace it with the issue of judgement.
>
> —Brian Eno

The extent to which technique is a defensive moat is the extent to which human expression is being limited by available tools. Collapse the distance between imagination and implementation by building tools.

Every time a tool bridges a defensive moat, there will be arguments about what counts as "real art".

- Where is techique a defensive moat?
- Where do you see "ameteur" or DIY subcultures? Example: Homebrew Computer Club, DIY, Maker Movement, Indie Biology. What tools are they missing? What tools are inaccessible to them? How can you create **access to tools**?
- What kinds of experience cannot be replaced or unlocked via a tool? Where does replacing technique with a tool cause harm?

You'll need **access to tools**, and you may want to consider **bootstrapping**.

Backlinks

Recreate a piece of art
Brick pencil
Human scale tools

Related

Amara's Law
Top 1000 tools
Human scale tools
Brick pencil
Scenius
Measure it
OODA loop
Creatives are measu…
Gall's Law
Thing from the Future
Bootstrapping
Conformability

Ecology meta-patterns

Emergent patterns that recur in complex adaptive systems:

- Diversity
- Recirculation
- Niche and hierarchy
- Coevolution

Diversity: an ecology is an economy made up of many currencies. Different species have different goals, and those goals interlock in nonlinear ways.

Recirculation: energy in the system is recycled and spent over and over again. Plants absorb sunlight, animals eat plants, animals fertalize plants. Over time, the ecology evolves complex recirculation webs.

Niche and hierarchy: Like Adam Smith's pin factory, when generalist **agents** can **evolve** into specialists, it increases the total efficiency and carrying capacity of the ecosystem. Specialization facilitates recirculation. Hierarchies of specialists emerge, and resources flow from specialist to specialist, as in an assembly line.

Coevolution: several relational tropes continually recur between specialists. Predator-prey, mutialism, commensalism, parasitism.

(Insights from Signals and Boundaries, MIT Press).

Evolution

Evolution is a behavior that emerges in any system that has:
Mutation Heredity Selection
Evolutionary systems often generate unexpected solutions

Punctuated equilibrium

Complex systems exist in punctuated equilibrium, repeatedly evolving through distinct phases of randomness, growth, consolidation, and collapse

Provoking emergence

So how do you provoke emergence? It's a process of gardening, not architecture

---

Related

Generative Grammar
Composable alphabets
Building Block Hypot…
Agent
Markets do not make…
Verbs that can act on…
Capping downside di…
Punctuated equilibrium
Gall's Law
Low floor, wide walls,…
Language is a Lamar…
Evolution is a diversif…

Punctuated equilibrium

Complex systems exist in punctuated equilibrium, repeatedly **evolving** through distinct phases of randomness, growth, consolidation, and collapse.

(Phase 1) Random: The system is unstructured. Random events occur without particularly changing the structure.

(Phase 2) Growth: An innovation causes a major phase transition within the structure of the system. The innovation catalyzes other innovations in a positive feedback loop.

(Phase 3) Consolidation: Growth rates saturate. The ecosystem consolidates into a highly organized network, optimized for efficiency, as each **agent** seeks to eke out as much as it can from its position in the value chain. Hubs (keystone species) appear at critical points.

(Phase 4) Collapse: A random shock, or new innovation demolishes one of the keystone species, causing cascade failure within the highly structured network. The ecosystem collapses into a random structure.

(Repeat): The system begins a slow crawl back up the **evolutionary ladder** toward a new metastable state.

Insights from the work of Sanjay Jain, Sandeep Krishna:

- Large extinctions in an evolutionary model: The role of innovation and keystone species
- A model for the emergence of cooperation, interdependence, and structure in evolving networks

Backlinks

TIMN

Ecology meta-patterns

Related

Generative Grammar
Extrastatecraft
Building Block Hypot…
Low floor, wide walls,…
Capping downside di…
Composable alphabets
Alphabets
Agent
Requisite Variety
Evolution is a diversif…
Genetic algorithms
Conformability

A Pattern Language

In A Pattern Language, Alexander argues for the primacy of **design patterns** as a language for creation, against top-down theory.

What is a design pattern? A template for a solving problem that has evolved through bottom-up emergence. Often patterns do not belong to an articulated theory — they are just the way things are done.

Ideology and theory are poor models for complex systems. They don't model the second-, third-, fourth-order effects of design decisions. By contrast, a design pattern is found embedded in a real-world complex system. It is an existance-proof of what can work in such a system.

Alexander argues for the primacy of pattern in architecture and city planning, but his influence has probably been greatest in the software engineering community, where design pattern has become a term of art.

Key takeaway
If patterns are the alphabet for systems design, then you can craft designs that work by collecting, understanding and combining patterns.

A good design pattern

- Has survived the test of time.

- Is the product of an evolutionary process of trial and error, rather than theory or ideology.
- Is found, not deduced. We should seek out patterns by observing thriving cities, towns, neighborhoods.

The book
The book is a catalog of interconnected, cross-referenced patterns at increasing levels of scale:

- The home
- The neighborhood (including transportation and infrastructure)
- The city
- The region

Where do patterns come from?
Alexander seems to think there is a through-line at the heart of all design patterns — a Zen-like property that can't be articulated through theory. He calls it The Quality Without a Name.

> There is a central quality which is the root criterion of life and spirit in a man, a town, a building, or a wilderness. This quality is objective and precise, but it cannot be named.

A Pattern Language is actually the second of 3 books. In The Timeless Way of Building, Alexander attempts to articulate this Quality Without a Name. This was followed by A Pattern Language, where Alexander's goal is to point to this Quality by cataloging patterns that display it. Finally, The Oregon Experiment documents Alexander's attempt to put these ideas into practice while planning the University of Oregon campus.

Backlinks

Human scale

Related

Soft security
Low floor, wide walls,…
Slowly, then all at once
Gall's Law
Capture, organize, s…
Gimmick Book
Thing from the Future
Games with more th…
Generative Grammar
Building Block Hypot…
Recreate a piece of art
Build to think

3 is the magic number

When **storytelling**, 3 is the minimum number of samples necessary to trigger pattern recognition. 1 is all by itself, 2 might be a coincidence, but 3 is a pattern.

Also **intuition is pattern recognition**.

Related

Ma
Three-act story struc…
In medias res
OODA loop
Create choices, mak…
Meaning, detail, drama
Brick pencil
Aspect-to-aspect
Structure by plot
Architects vs garden…
Human scale
What is, what could …

Start with a spreadsheet

Start with a spreadsheet. Put off writing an app for as long as possible. Spreadsheets are enough to support ~100 person efforts. Structure it like a database (columns are types, rows are records).

Backlinks

Judo move

Related

Modularize later
Start with a toy
Prototyping
Value, scale, time
100 Users
Charlie Kindel's 5 Ps

Architects vs gardeners

This idea was coined by Brian Eno in Composers as Gardeners.

> An architect, at least in the traditional sense, is somebody who has an in-detail concept of the final result in their head, and their task is to control the rest of nature sufficiently to get that built. Nature being things like bricks and sites and builders and so on. Everything outside has to be subject to an effort of control.
>
> A gardener doesn't really work like that. Unless it's, as Mark's mentioned, Versailles, which is, to me, the most grotesque of all gardens, since it's the total denial of nature and the complete expression of human control over nature. So it's a perfect forerunner to the Industrial Age, Versailles. But what I think about, I suppose my feeling about gardening, and I suppose most people's feeling about gardening now, is that what one is doing is working in collaboration with the complex and unpredictable processes of nature. And trying to insert into that some inputs that will take advantage of those processes, and as Stafford Beers said, take you in the direction that you wanted to go.
>
> Use the dynamics of the system to take you in the direction you wanted to go. So my feeling

has been that the whole concept of how things are created and organized has been shifting for the last 40 or 50 years, and as I said, this sequence of science as cybernetics, catastrophe theory, chaos theory and complexity theory, are really all ways of us trying to get used to this idea that we have to stop thinking of top-down control as being the only way in which things could be made.

We have to actually lose the idea of intelligent design, because that's actually what that is. The top-down theory is the same as intelligent design. And we have to actually stop thinking like that and start understanding that complexity can arise in another way and variety and intelligence and so on. So my own response to this has been, as an artist, to start to think of my work, too, as a form of gardening. So about 20 years ago I came up with this idea, this term, 'generative music,' which is a general term I use to cover not only the stuff that I do, but the kind of stuff that Reich is doing, and Terry Riley and lots and lots of other composers have been doing.

> Since I have always preferred making plans to executing them, I have gravitated towards situations and systems that, once set into operation, could create music with little or no intervention on my part.
>
> That is to say, I tend towards the roles of planner and programmer, and then become an audience to the results.

— Brian Eno, Music, time and Long-term thinking

George RR Martin (Game of Thrones):

> I've always said there are – to oversimplify it – two kinds of writers. There are architects and gardeners. The architects do blueprints before they drive the first nail, they design the entire house, where the pipes are running, and how many rooms there are going to be, how high the roof will be. But the gardeners just dig a hole and plant the seed and see what comes up. I think all writers are partly architects and partly gardeners, but they tend to one side or another, and I am definitely more of a gardener. In my Hollywood years when everything does work on outlines, I had to put on my architect's clothes and pretend to be an architect. But my natural inclinations, the way I work, is to give my characters the head and to follow them.

A gardening approach is well-suited to **provoking emergence**.

Storytelling

Techniques for telling stories well:
The traditional story arc

Backlinks

Procedural paintbrush Permissionless creat… General-purpose lan…

Related

Value, scale, time

> Community gardens, time banks, credit unions, etc are small, but they're economic institutions that are resilient, sustainable, ppl-centered
>
> — Mai Sutton (@maira) August 17, 2016

Often, we associate impact with bigness, speed: large-scale interventions, disruptive innovations, fast-growing startups, the iPhone, solar energy projects, that kind of thing.

Another way to think about impact is to include the time variable:

impact = (value * scale) * time
A neighborhood park that lasts genenerations, a tree planted out front, a library. Small things that last a long time have high impact.

Fundamental Needs

Fundamental needs are universal constraints

Related

Start with a spreadsh…
Goodhart's law
Gimmick Book
Games with more th…
Bootstrapping
Prototyping

Measure it
Slowly, then all at once
Gall's Law
Tools over technique
OODA loop
Design patterns

OODA loop

- Observe
- Orient
- Decide
- Act

A strategic thinking framework developed by Colonel John Boyd, USAF. Since the theater of war is constantly changing, OODA treats decision-making as a control theory problem. You're constantly measuring and readjusting to trend toward your goal — which might be moving.

OODA values agility as a strategic advantage.

Questions:

- What phase of the OODA loop am I in now? Refocus tactics on that phase.
- What are my go-to-tools for observation? What tools do others use?
- How do I orient when faced with a new problem space?
- Is it better to make decisions quickly or carefully? OODA encourages rapid measure-decide-act loops.

> To understand is to know what to do.
>
> - Ludwig Wittgenstein

And also: **solvitur ambulando**.

More:

- OODA on Wikipedia

Backlinks

Harvest what you plant

Related

Intuition is pattern re…	Design patterns	Generalized problem…
Gall's Law	Tools over technique	Write the press release
4 ways to augment t…	Learn to learn	Start with a toy
Slowly, then all at once	3 is the magic number	Job-to-be-done

Papers are more durable than programs

Papers are more durable than programs. Think Mozart — we have his sheet music, but not his implementation.

Concept from How to write a great research paper (Peyton-Jones).

Backlinks

Top 1000 scientific p…

Related

Headings are tweets
Top 1000 scientific p…
Top 1000 tools
LOCKSS

Judo move

Judo move: maximum result, minimum effort.

- Reframing the problem so it is no longer a problem.
- The simplest thing that could possibly work.
- The decision that massively reduces the scale of the problem.
- Moving with the momentum of the problem. Embracing its strengths and limits.

> Don't oppose forces, use them.
> —Bucky Fuller

Start with a toy, **start with a spreadsheet**. Do the effortless things that get you 80% and build momentum.

> Don't be afraid of things because they're easy to do. —Oblique Strategies

---

Hat tip 37 Signals.

Related

Architects vs garden…
Conversational UI
Human scale tools
Brainstorming
Brick pencil
Second System Syn…

Invisible Cities
Desirability, feasibilit…
Build to think
Jack principles
Gimmick Book
Modularize later

Recreate a piece of art

If you want to learn to paint like da Vinci, why not learn from da Vinci? If you want to learn to sculpt like Michelangelo, why not learn from Michelangelo?

Pick a piece of art. Try to recreate it in detail. Reverse-engeineer the techniques used to create it. Fail. Try again. Repeat.

Imitate art for practice.

On the other hand — **tools over technique**.

Related

Solvitur ambulando
A Pattern Language
Augmented convers…
Intuition is pattern re…
Pensees
Low floor, wide walls,…
Build to think
The Art of Game Des…
Invisible Cities
Brick pencil
3 is the magic number
Understanding Comics