Labso Pattern Language

From Appreciating Systems Wiki

Contents

Principles

Source of Christopher Alexander's Pattern Language: https://archive.org/details/APatternLanguage. Worth noting also are Ward Cunningham's Tips (http://www.c2.com/cgi/wiki?TipsForWritingPatternLanguages)

Just below are the underlying principles that inspire me Nicolas Stampf (admin) to create the LAbso's Design Patterns

Those of Appreciative Inquiry:

- what you focus on, grows
- you go toward what you repeatedly inquire about
- the more positive the inquiry, the more attractive it gets
- change begins the moment you ask the first question

Those of Solution Focus:

- In every broken system, something works: you only have to find it out - whatever it is you're looking for, it's already happening in the system to some extent (maybe only partially): you only have to find it out - the smallest step's the better - the solution is IN the interaction

Or SIMPLE principles as advocated in Paul Z Jackson & Marl MCKergow book (http://www.thesolutionsfocus.com/):

1. S-olutions, not problems;
2. I-nbetween — the action is in the interaction;
3. M-ake use of what's there;
4. P-ossibilities — past, present and future;
5. L-anguage — simply said; and
6. E-very case is different.

Those of Positive Deviance:

As per the PD guide (http://www.positivedeviance.org/about_pd /Final%20revised%2006-03-09Basic_PD_Steps_2.pdf):

Guiding Principles of Positive Deviance Approach :

1. **Community** discovers existing, uncommon, successful behaviors and strategies (PD Inquiry) and unleashes great ideas
2. **Community** creates plans to amplify successful practices and act on ideas
3. **Community** recognizes that "someone just like me" can do and get better results especially in the worst case scenarios (social proof).
4. Emphasis is on transferring *behavior* instead of knowledge. "Act your way into a new way of thinking instead of thinking your way into a new way of acting"
5. **Community** owns the entire process "Everyone" is involved – Go to uncommon places and to "unlikely suspects to find solutions. "Don't do anything about me without me"
6. **Community** –creates its own (actionable) performance indicators and monitors progress.

Clearly, the force of the PD resides in the **Community** doing all of the work.

Grammar

Exemple / syntax

Source: Wikipedia http://en.wikipedia.org /wiki/Pattern_language#Simple_example_of_a_pattern. More comprehensive format: http://www.designmatrix.com/pl/anatomy.html

- **Name**: ChocolateChipRatio
- **Context**: You are baking chocolate chip cookies in small batches for family and friends
- **Consider these patterns first**: SugarRatio, FlourRatio, EggRatio
- **Problem**: Determine the optimum ratio of chocolate chips to cookie dough
- **Solution**: Observe that most people consider chocolate to be the best part of the chocolate chip cookie. Also observe that too much chocolate may prevent the cookie from holding together, decreasing its appeal. Since you are cooking in small batches, cost is not a consideration. Therefore, use the maximum amount of chocolate chips that results in a really sturdy cookie.
- **Consider next**: NutRatio or CookingTime or FreezingMethod

Patterns

I (Nicolas Stampf (admin)) think it would probably be better to articulate these patterns around the 2 basic elements of a Labso (strengths and social), which can be separated in four (VIES (french for lives): Visions (of where you want to go), Ideas (of how to get there), Experiences (successful ones going toward your visions) and Social (used to help in identifying, amplifying and refracting the preceding three strengths)). To facilitate a LAbso is to awake all four. Then for each one of them, we can face the same difficulties:

- no idea: see in the past what's been good so far (Discovery phase of AI)
- don't know what to do

1. 1. look in the past or present for when the person indeed was able to do, at least partially (solution focus)
   2. the ponder how the person herself contributed to make that (even small) success

- dig the problem
  - either ask what is wanted instead of it (positive transformation of AI) - neutral situation
    - and push further by asking "what would be even better than 'netural situation'?"
  - or ask when is the problem absent (SF)
- doesn't know where to start

1. 1. ensure the vision is clear enough to begin with
   2. do a retroplanning (possibly big changes / provocative propositions of AI)
   3. ask for the first smaller steps (solution focus) to begin the change

Prevent Resistance

- **Context**: general
- **Consider these patterns first**:

- **Problem**: people are resisting change “NIH”
- **Solution**: don’t propose anything. Ask questions only about what they want, and start from that.
- **Consider these patterns next**:

Vision

- **Context**: you are working the vision toward which people will change
- **Consider these patterns first**:
- **Problem**: people don’t know what attracts them
- **Solution**: Ask people about a time where they were happy and feeling like they were doing something meaningful to them. Ask why did it made sense. Ask why again (9 times: Liberating

Structures “9 whys”)

- **Consider these patterns next**:

Finding what works

- **Context**: you are seeking solutions to replicate/amplify/adapt
- **Consider these patterns first**:
- **Problem**: faced with a problem, you need a solution
- **Solution**: ask about when the problem is/was absent, and what prevented it.
- **Consider these patterns next**:

Stop investigating a problem

- **Context**: In the flow of conversation, someone starts wondering about what might be causing the

problem so as to analyze the cause in order to prevent it

- **Consider these patterns first**:
- **Problem**: Conversation starts drifting toward problem analysis
- **Solution**: Ask about what people want instead of the problem. Ask about when was the problem absent and how things were instead.
- **Consider these patterns next**:

Find solutions elsewhere

- **Context**: you’re looking for solutions, but people can’t seem to find ones
- **Consider these patterns first**: Finding what works
- **Problem**: Despite using “finding what works”, people can’t find solutions.
- **Solution**: Ask about someone else that might be in a similar situation and still manage to handle it satisfactorily (or with the problem being of a smaller magnitude). Then go see and talk with this person to find out what she does that prevents the problem to some extent.
- **Consider these patterns next**:

Build a path toward a solution

- **Context**: Having found some solutions, you need to act to apply them
- **Consider these patterns first**:
- **Problem**: People don’t know how to start doing the solution
- **Solution**: Ask about what does the situation looks like when the solution will be in place. Then ask about what will be different. What else? Build as rich a picture as possible. Then ask about what needs to have change to allow all these different things to be. Then what needed to be done to allow them. Then what before, etc. Retroconstruct step by step.
- **Consider these patterns next**: Kickstart a solution

Kickstart a solution

- **Context**: a solution has been identified, broad directions set
- **Consider these patterns first**:
- **Problem**: People don’t know where and how to start
- **Solution**: Ask to refine the directions and detail each action. Then, for each action, ask about what small, physical, concrete step needs to be done to begin it. Then a smaller step. Then a even smaller. Again.
- **Consider these patterns next**:

Finding confidence on our own skills

- **Context**: problems, solutions, strengths, ideas have been shared in a group, ideas and pairs of people are connected
- **Consider these patterns first**:
- **Problem**: some people may feel a lack of confidence in their own skills to accomplish their solution / action
- **Solution**: ask to their pairs why they know that they will succeed
- **Consider these patterns next**:

Connect ideas

- **Context**: a brainstorming has been done, and full of ideas are on the wall
- **Consider these patterns first**:
- **Problem**: can't connect ideas together and form pairs
- **Solution**: in front of the wall, ask people to understand others ideas, ask them to find their preferred idea except theirs, ask them to discuss with the author of the idea
- **Consider these patterns next**:

Connect out of the labso

- **Context**: a labso has ran, stories have been shared, ideas and solutions have been found
- **Consider these patterns first**:
- **Problem**: people may not connect with pairs outside labso without a nice invitation

- **Solution**: ask what really engage each individuals on their next steps, ask them to connect by pair to help each other and work on their preferred topics. Ask them to fix a first rendez-vous before leaving the labso.
- **Consider these patterns next**:

Build on differences

- **Context**: People exchange ideas, and everybody's ideas are different one from another.
- **Consider these patterns first**:
- **Problem**: The usual step is to focus on the differences and find arguments why one idea might be better than another.
- **Solution**: Some principles can help use the differences to grow a solution rather than a fight: 1) Don't assume one idea is better than another. Seek to understand the reasons behind another idea. Assume you lack information and ask the other person why is she promoting that idea. (In David Bohm's language, we talk of suspending judgment and "balancing inquiry and advocacy (https://www.solonline.org/?page=Tool_InquiryAdvocacy)". 2) However irreconcilable the ideas may seem, do seek the common ground between them: find what connects them, what is similar in both of them. Then identify what each idea allows or brings as a consequence, and, frmo the bases, try to construct a new idea that would provide BOTH consequences at best, or difference yet acceptable consequences to both holders.
- **Consider these patterns next**:

Learn to ask questions to others

- **Context**:
- **Consider these patterns first**:
- **Problem**:
- **Solution**:
- **Consider these patterns next**:

ANNEX Don't use the code below, it's the template! :-)

Title

- **Context**:
- **Consider these patterns first**:
- **Problem**:
- **Solution**:
- **Consider these patterns next**:

/index.php?title=Labso_Pattern_Language&oldid=3570"

Category: Labso

- This page was last modified on 26 May 2015, at 14:08.